



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

VIKAKOODINLUKIJA

Mercedes-Benz

TEKIJÄ/T: Erno Hyvönen

Koulutusala Tekniikan ja liikenteen ala			
Koulutusohjelma Automaatiotekniikan koulutusohjelma			
Työn tekijä(t) Erno Hyvönen			
Työn nimi Vikakoodinlukija			
Päiväys	29.5.2013	Sivumäärä/Liitteet	40/3
Ohjaaja(t) Risto Niemi			
Toimeksiantaja/Yhteistyökumppani(t)			
<p>Tiivistelmä</p> <p>Tämä insinöörityö tehtiin freelancer–tyylisesti ilman yhteistyökumppaneita. Työn tavoitteena oli suunnitella ja toteuttaa mikrokontrolleripohjainen vikakoodinlukija vuodesta 1980 vuoteen 1994 valmistettuihin Mercedes-Benz henkilöautomalleihin W126, W123, W201 ja W124. Vikakoodinlukijan tärkeimpänä lähtökohtana oli helppokäyttöisyys ja kestävyys autokäytön vaativissa olosuhteissa.</p> <p>Työssä suunniteltu ja toteutettu vikakoodinlukija lukee pulssi- ja PWM-muotoisia väyliä. Lisäksi laitteella voidaan tyhjentää pulssimuotoiset koodit muistista sekä lukea auton moottorinohjausjärjestelmän reaaliaikaisia arvoja, jotka ovat jännite, kierrosluku ja lambda-anturin säätöarvot.</p> <p>Laitteen elektroniikka ja piirilevy on suunniteltu National Instruments Multisimilla ja Ultiboardilla. Piirilevy on valmistettu ProtoMat C30/s-piirilevyjyrsimellä ja siihen liittyvillä ohjelmilla CircuitCam ja BoardMaster. Kontrollerin ohjelmoinnissa on käytetty Matrixin Flowcodea ja MicroPro-ohjelmaa.</p> <p>Laitteen käyttöliittymä suunniteltiin helppokäyttöiseksi. Vikakoodinlukijaan on ohjelmoitu opas, joka neuvoa ta-pauskohtaisesti johtojen kytkennän väylän liittimiin. Opas myös kertoo pitääkö sytytysvirran olla päällä vai moottorin käynnissä. Laitteessa on vain kuusi painonappia. Valikot ja painikkeiden tarkoitus näkyvät laitteen näytöllä.</p>			
Avainsanat Vikakoodinlukija, kontrolleri, Mercedes-Benz, pulssi, PWM, pulssisuhde			

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Automation Technology			
Author(s) Erno Hyvönen			
Title of Thesis Fault Code Scanner			
Date	29.5.2013	Pages/Appendices	40/3
Supervisor(s) Risto Niemi			
Client Organisation /Partners			
<p>Abstract</p> <p>This thesis work was made as a freelancer without any affiliates. The goal of the thesis was to design and execute a microcontroller-based fault code scanner for Mercedes-Benz models W126, W123, W201 and W124 made in 1980-1994. The main goal was to create foolproof and durable device which can last in demanding circumstances.</p> <p>Designed and built fault code scanner reads pulse- and PWM –shaped buses. Device is also capable of clearing saved pulse-shaped fault codes and reading realtime codes such as voltage, revolutions per minute and lambda-sensor’s adjustment value from engine’s electronic control unit.</p> <p>The electronics and circuitboard was planned by National Instruments Multisim– and Ultiboard –programs. The circuit board is made by ProtoMat C30/s PCB printer with programs CircuitCam and BoardMaster. Microcontroller is programmed with Flowcode and MicroPro.</p> <p>The user interface was planned easily accessible. There is a guide programmed in to the microcontroller which tells how to connect wires to the connectors. It also tells in which state the engine must be, ignition on or engine running. The fault code scanner has just six buttons. The device shows menus and buttons purposes in the LCD–screen.</p>			
<p>Keywords</p> <p>Fault code scanner, microcontroller, Mercedes-Benz, pulse, PWM, duty cycle</p>			

SISÄLTÖ

1	JOHDANTO	8
2	VIKAKOODINLUKIJAJA	9
2.1	Väylät	9
2.1.1	Pulssimuotoinen väylä	9
2.1.2	Pulssisuhde -väylä	10
3	MIKROKONTROLLERIT	11
4	ELEKTRONIIKAN KOMPONENTIT	13
4.1	Passiiviset	13
4.1.1	Vastukset	13
4.1.2	Kondensaattorit	14
4.1.3	Diodit	14
4.2	Aktiiviset	15
4.2.1	Transistorit	15
4.2.2	Regulaattorit	15
4.2.3	Kiteet	16
4.3	Muut	16
5	LAITTEET JA OHJELMAT	17
5.1	Matrix Flowcode	18
5.2	National Instruments Multisim	18
5.3	National Instruments Ultiboard	18
5.4	CircuitCam 4.0	18
5.5	BoardMaster 4.0	19
5.6	Protomat	19
6	KYTKENTÄKAAVIO	19
6.1	Piirikaavion analysointi	19
7	OHJELMA	23
7.1	Pulssimuotoinen väylä	24
7.2	Pulssisuhde ja live-data	25
8	TYÖN TOTEUTUS	27
8.1	Kytkenän piirtäminen	27
8.2	Piirilevyn suunnittelu	28

8.3	Piirilevyn valmistus.....	30
8.4	Juottaminen	33
8.5	Ohjelma	33
9	KOTELOINTI	35
9.1	Kotelon työstö	36
9.2	Pintakäsittely	37
10	PÄÄTELMÄT	39

LYHENTEET JA MÄÄRITELMÄT

4Matic = Mercedes – Benzin suunnittelema ja käyttämä nelivetojärjestelmä

AC = Air Conditioning

ASD = Automatic Slip Differential

CAN = Controller Area Network

CIS-E = Continuous fuel Injection System Electric

EDS = Electronic Diesel System

ELR = Diesel injection system – electronic idle speed control

Footprint = komponentin jalkajärjestys

HVP = High Voltage Programming

ICSP = In Circuit Serial Programming

LCD = Liquid Crystal Display

LVP = Low Voltage Programming

OBD = On Board Diagnostics

RAM = Random Access Memory

RPM = Rotations Per Minute

SD = Self-Diagnosis

SRS = Supplemental Restraint System

Mielenkiinto vikakoodinlukijan hyödyntämiseen syntyi autoharrastuksen parissa. Käytännön tarve vikakoodinlukijalle tuli erään W201 2.0E Mercedes-Benzin hankalien sähkövikojen ja siitä seuranneiden käyntihäiriöiden vuoksi. Markkinoilla ei kuitenkaan ollut mitään sopivaa laitetta tarjolla, johtuen tuohon aikaan valmistajien käyttämistä omista diagnostiikkaprotokollista, joten asia jäi sillä erää siihen. Kevätlukukaudella 2012 pidetyllä kenttäväylät ja -laitteet kurssilla piti perehtyä johonkin väylään ja hyödyntää sitä jossain laitteessa. Kehittelin tällöin vikakoodinlukijaa ajatustasolla, mutta käytännön sovellusta siitä ei tullut. Laitteen suunnittelu ja toteuttaminen osoittautuivat osaamisen suhteen ylivoimaiseksi. Nyt vikakoodinlukijan toteutukseen tarjoutui insinööriyön muodossa hyvä tilaisuus ja saatuani aihekuvauksen hyväksytyksi aloitin työn teon.

Aihe on rajattu rajaamalla vikakoodinlukijan sopivuus vain Mercedes-Benz-malleihin W126, W123, W201 ja W124. Useammat tuetut mallit tekisivät ohjelmasta liian monimutkaisen ja johtoja väyliin pitäisi olla monta erilaista. Myöskään aika annetuissa puitteissa ei riittäisi tällaisen laitteen suunnitteluun ja toteuttamiseen.

Vikakoodinlukijan suunnittelun lähtökohtana on helppokäyttöisen ja kestäväen laitteen luominen vaativiin olosuhteisiin.

Haluan kiittää Esedun Pieksämäen henkilökunnasta lehtori Raimo Kuusistoa ja lehtori Aku Hyvöstä yhteistyöstä piirilevyn valmistamiseen liittyvien laitteiden järjestämisestä. Haluan kiittää myös yliopettaja Antti Ruotsalaista ja laboratorioinsinööri Tapio Kiantoa ammattitaitoisesta ja mielenkiintoisesta opetuksesta. Ilman heitä en olisi tätä insinööriyötä kyennyt toteuttamaan itsenäisesti. Kiitos kuuluu myös opinnäytetyön ohjaajille lehtori Risto Niemelle ja lehtori Markku Halttuselle mutkattomasta yhteistyöstä insinööriyön puitteissa.

2 VIKAKOODINLUKIJJA

Vikakoodinlukija on laite, jolla voidaan purkaa auton diagnostiikkajärjestelmään tallentuneet koodit. Laitteella voidaan lukea tallentuneet koodit ihmisen ymmärtämässä muodossa eli tekstinä tai numeroina. Numeroiden tapauksessa koodeja verrataan taulukkoon, josta saadaan selitykset koodeille. Osassa vikakoodinlukijoista on myös live data-arvot saatavilla. Nämä ovat reaaliaikaisia arvoja auton antureista ja toiminnoista. Tällaisia ovat esimerkiksi kierrosluku ja jännite.

Vikakoodinlukija on oivallinen apuväline autoista kiinnostuneelle ja auton huoltoihin vihkiytyneelle harrastajalle. Varsinkin pohjoisissa olosuhteissa, nykyajan monimutkaisilla ja siten vikaherkillä sähköjärjestelmillä varustettuun autoon se on elintärkeä. Monet viat eivät enää ole loogisia ja saattavat vaatia kunnostuksen jälkeen vikakoodien nollauksen toimiakseen. Osien vaihtaminen haupitsityyliin ei siis ole järkevää saati halpaa eikä vika välttämättä poistu, vaikka viallinen osa tulisivat vaihdetuksi.

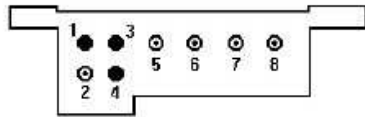
Edulliset vikakoodinlukijat ovat yleistyneet markkinoilla hintahaitarin ollessa 30-100 euroa. Halvimmissa laitteissa ei kuitenkaan välttämättä ole kaikkia ominaisuuksia mitä kalliimmissa, kuten live datan näyttöä. Joka tapauksessa, varsinkin CAN-väyläisen auton omistaja voi säästää satoja euroja vuodessa hankkimalla oman vikakoodinlukijan.

2.1 Väylät

Väylät voidaan jakaa kahteen eri ryhmään, nopeisiin ja hitaisiin väyliin. Jako nopeiden ja hitaiden väylien välillä tehdään pulssitiheyden perusteella (WHITE, Charles 1998). Mikäli ihmissilmä erottaa pulssit vaikkapa väylään kytketystä LEDistä, on kyseessä hidas väylä, muutoin nopea. Väylien signaali on joko nolla volttia tai järjestelmän jännite (sytytysvirta päällä 12,7 volttia tai auto käynnissä 14,4 volttia). Tässä kappaleessa syvennytään työssä käytettyjen diagnostiikkaväylien toimintaan.

2.1.1 Pulssimuotoinen väylä

Mercedes-Benzin hitaisiin väyliin lukeutuvasta (WHITE, Charles 1998) pulssimuotoisesta väylästä saa autoon kytkettyjen lisävarusteiden koodit. Jokaiselle laitteelle on oma väylänsä ja vikakoodin lukeminen alkaa maadoittamalla väylä sekunniksi, jonka jälkeen koodi tulee pulsseina. Ennen pulssikoodin alkamista kuluu muutama sekunti (WHITE, Charles 1998). Pulssien lukumäärä verrataan taulukkoon (liite 2), josta saadaan selville koodin aiheuttaja selville. Vikakoodit tallentuvat muistiin, joten ne voidaan nollata vetämällä kyseinen väylä maihin kahdeksaksi sekunniksi. Väylän liitin sijaitsee moottoritilassa oikealla puolella akun läheisyydessä.



8-pole Diagnostic Connector

Models 201, 124, 126

1	Ground
2	Not used
3	CIS-E Continuous fuel injection system (CFI)
4	ELR Diesel injection system - Electronic idle speed control system
5	EDS Electronic diesel system
6	ASD Automatic locking differential
7	4MATIC Automatic-engaged four wheel drive (124 only)
8	SRS Supplemental Restraint System
	A/C Air Conditioning
	Not used

Kuva 1. Pulssimuotoisen väylän liitin (MBCluster)



Kuva 2. Liitin W201-korisen Mercedes-Benzin moottoritilassa

2.1.2 Pulssisuhde -väylä

Ruiskutusjärjestelmällä varustettuun bensiinimoottoriin on kytketty 9-napainen liitin. Tästä liittimestä saadaan reaaliaikaisia moottorin arvoja, joita ovat kierrosluku, sytytysennakko, lambdan säätöarvo ja pulssisuhde-muotoinen vikakoodiväylä. Pulssisuhde saadaan vertaamalla signaalin alhaaloloaika ylhäälläloolokaan. Vikakoodille on selitys 10 prosentin välein. Opinnäytetyössä ei kuitenkaan ole vertailutaulukkoa, sillä en löytänyt luotettavaa lähdettä niihin. Väylän liittämä sijaitsee vasemmalla puolella moottoritilassa.

Taulukko 1. X11-liittimen pinout

1	RPM
2	GND
3	Fault Code Line
4	Low side of ignition coil
5	Hot in run or start
6	Hot at all times
7	TDC sensor
8	Shielded cable from TDC
9	Shield



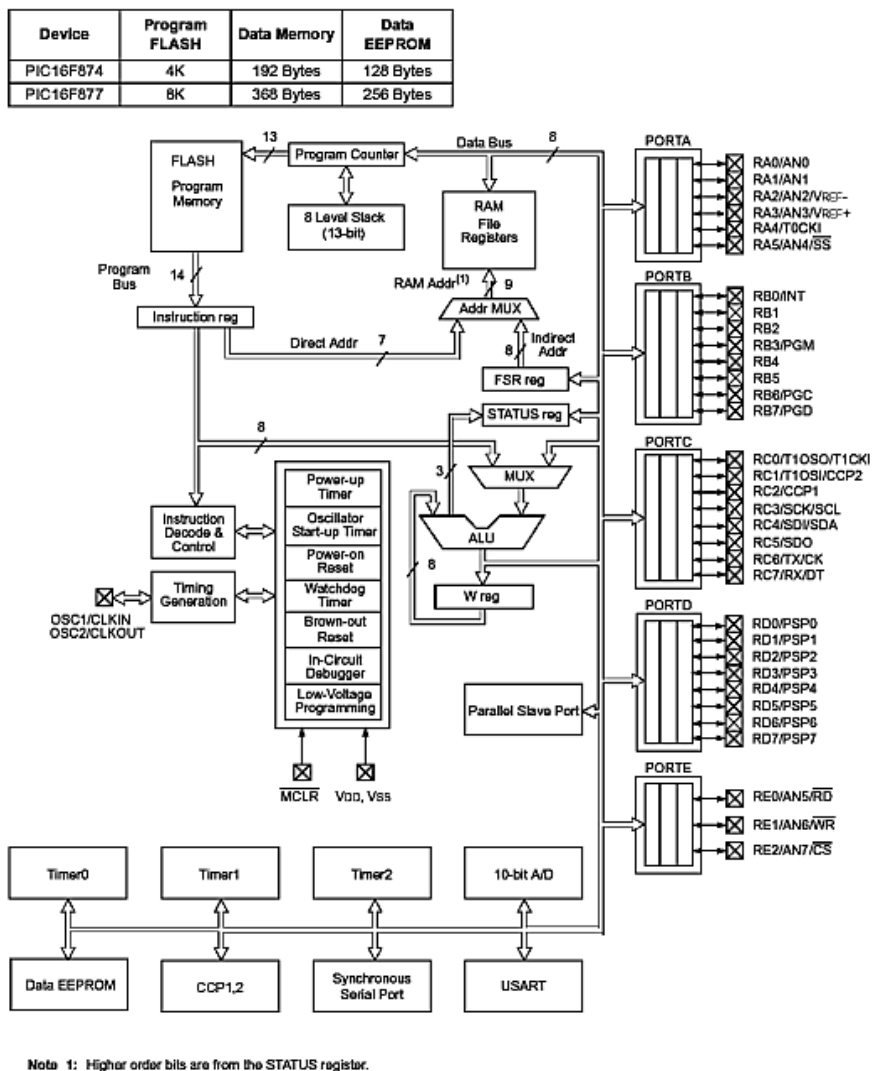
Kuva 3. X11-liitin W201-mallin Mercedes-Benzissä.

3 MIKROKONTROLLERIT

Mikrokontrolleri on monille tuntematon käsite, vaikka jokainen käyttää päivittäin kontrolleripohjaisia laitteita. Tällaisia ovat esimerkiksi tietokoneen hiiri, kännykkä, kaukosäädin ja monet auton sähkölaitteet. PIC-mikrokontrolleriin olen törmännyt W201-korimallin bensiinipumpun releen yhteydessä.

Työssä käytetyn Microchip PIC:n lisäksi on muitakin kontrollerin valmistajia ja malleja. Suosituimpia ovat AVR:n valmistamat kontrollerit, jota käytetään myös Arduino-ympäristössä. AVR:n etu muihin valmistajiin on laajat, ilmaiset kehitysyökalut ja -ympäristöt sekä suuri käyttäjäjoukko, joten apua ongelmiin löytyy aiheeseen keskittyviltä keskustelupalstoilta.

3.1 Kontrollerin rakenne



Kuva 4. 16F877A-kontrollerin sisäinen rakenne (Microchip).

Kuvasta 4 nähdään kontrollerin olevan 8-bittinen sen käyttämästä sisäisestä dataväylästä johtuen. Kontrolleriin ajettu ohjelma tallentuu sähköisesti tyhjennettävään FLASH-muistiin. Muuttujien arvot tallennetaan RAM-muistiin, jolloin virtojen katketessa niiden sisältämät arvot tyhjenevät.

3.2 Ohjelmointi

Mikrokontrolleri ohjelmoidaan HVP:llä nostamalla Vpp-tulo ylös 13 volttiin, jolloin laite menee ohjelmointitilaan. Tämän jälkeen syötetään kello- ja datasiinaali kontrolleriin. ICSP:llä ohjelmoitaessa on käytössä LVP-ohjelmointi, jolloin nostetaan B3-tulo ylös viiteen volttiin ja saavutetaan ohjelmointitila. Muutoin ohjelmointi on samankaltainen kuin HVP:tä käytettäessä. (Microchip)

4 ELEKTRONIIKAN KOMPONENTIT

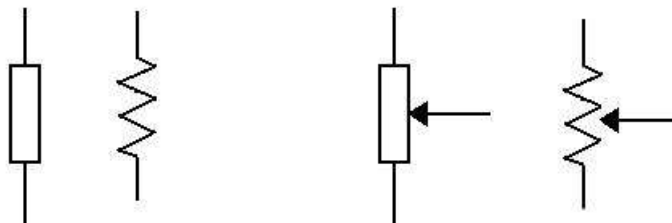
Vikakoodinlukijasta voidaan havaita kaksi pääosiota: elektroniikka ja ohjelma. Komponentit voidaan jakaa kahteen pääryhmään: aktiiviset ja passiiviset. Tässä kappaleessa kerrotaan lyhyesti käytännön tietoa työssä käytetyistä elektroniikan komponenteista.

4.1 Passiiviset

Vastukset, kondensaattorit ja kelat ovat passiivisia komponentteja, sillä ne eivät osallistu suureiden vahvistamiseen (HAIKO, Timo 2013). Tässä kappaleessa kerrotaan passiivisista komponenteista.

4.1.1 Vastukset

Vastus on elektroniikan yleisin käytetty komponentti. Vastuksen tehtävänä on rajoittaa virtaa ja jännitettä. Vastusten tärkeimmät suureet ovat resistanssi R [Ω] ohmeina, tehonkesto P [W] watteina ja toleranssi [%]. Yleisimmät vastustyytit ovat hiili- ja metallikalvovastukset, lankavastukset sekä massavastukset. Massavastuksiin törmää nykyään lähinnä vanhoissa laitteissa sekä musiikki-instrumenttivahvistimissa ja niiden sovelluksissa. On olemassa myös säädettäviä vastuksia, potentiometrit ja työkalulla säädettävät eli trimmerit. Vikatilanteessa vastus kuumenee ja palaa lopulta poikki. Kuvassa 5 esitetään vastusten ja säätövastusten piirrosmerkit.



Kuva 5. Vastusten piirrosmerkit

Vastuksien tunnistamiseksi on monta tapaa. Ne voidaan mitata, katsoa värikoodien perusteella tai ison tehonkeston vastuksissa sekä säätövastuksissa, lukea suoraan vastuksen kyljestä. Tässä työssä käytettyjen 0,25W tehonkeston hiilikalvovastuksissa on värikoodaus. Vastuksessa on neljä rengasta, joiden jokaisella värillä on tarkoituksensa kuten alla olevasta taulukosta 2 nähdään.

Taulukko 2. Vastusten värikoodit

Väri	1.	2.	kerroin	toleranssi
Musta	0	0	10^0	$\pm 1\%$

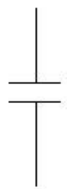
Ruskea	1	1	10^1	$\pm 2\%$
Punainen	2	2	10^2	
Oranssi	3	3	10^3	
Keltainen	4	4	10^4	
Vihreä	5	5	10^5	$\pm 0,5\%$
Sininen	6	6	10^6	$\pm 0,25\%$
Violetti	7	7	10^7	$\pm 0,1$
Harmaa	8	8	10^8	$\pm 0,05$
Valkoinen	9	9	10^9	
Kulta			0,1	± 5
Hopea			0,01	± 10
Tyhjä				± 20

Esimerkiksi vastus, jonka renkaat ovat ruskea, musta, oranssi ja kulta. Taulukkoa 2 vertaamalla saadaan $10 \times 10^3 \pm 5\%$ eli $10000\Omega \pm 5\%$ eli $10k\Omega \pm 5\%$. Kyseessä on siis 10 kilo-ohmin vastus, jossa 5% toleranssi. Näin ollen vastuksen koko on väliltä 9500-10500 ohmia.

Insinööriyössä vastuksia on hyödynnetty virran ja jännitteen rajoittamiseen sekä jännitteenjakokytkennöissä.

4.1.2 Kondensaattorit

Kondensaattori on komponentti, joka varastoi energiaa sen sisällä olevien levyjen väliin. Tärkeimmät kondensaattorin ominaisuudet ovat kapasitanssi C [F] Faradeina ja jännitteenkestoisuus U [V] voltteina. Kondensaattoria voidaan käyttää energiavarastona ja kytkentäkondensaattoreina sallien vain vaihtotaajuisten signaalien läpäisy. Yleisimmät kondensaattorityypit ovat elkot (elektrolyyttikondensaattorit), polkot (polyesterkondensaattorit) ja kerkot (keraamiset kondensaattorit). Kondensaattorin piirrosmerkeistä on monia variaatioita, joten kuvassa 6 esitetään yleinen kondensaattorin piirrosmerkki.



Kuva 6. Kondensaattorin piirrosmerkki

4.1.3 Diodit

Diodi on elektroniikan komponentti, joka läpäisee virran vain yhteen suuntaan: anodilta katodille. Diodi alkaa johtamaan, kun anodilla on diodin kynnyksjännitteen verran suurempi jännite kuin katodilla. Kynnyksjännite riippuu diodin materiaalista. Yleisimmät ovat piidiodi, jonka kynnyksjännite noin 0,7 voltia ja germaniumdiodi, jonka kynnyksjännite on noin 0,4 voltia. Diodissa katodi on merkitty viivalla ja diodin tyyppi on painettu koteloon. Vikatilanteissa diodi tuhoutuu palamalla oikosulkuun tai kokonaan poikki.



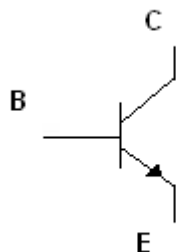
Kuva 7. Diodin piirrosmerkki

4.2 Aktiiviset

Aktiivisiksi komponenteiksi voidaan kutsua komponentteja, jotka vahvistavat jännitettä, virtaa tai tehoa (HAIKO, Timo 2013)

4.2.1 Transistorit

Transistoreja on kahta tyyppiä: NPN ja PNP. Insinööriyössä on käytetty yleisempää NPN –tyyppiä. Transistorin jalat ovat kollektori C, kanta B ja emitteri E. Transistori alkaa johtamaan kun kannalla on kynnyksjännitteen verran suurempi jännite kuin emitterillä ja kollektori-emitteri välin jännite suurempi kuin nolla voltia.

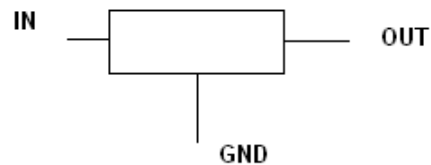


Kuva 8. Transistorin piirrosmerkki

4.2.2 Regulaattorit

Regulaattori on integroitu, kolmejalkainen komponentti, joka sisältää regulaattorikytkennässä käytetyn zenerdiodin ja transistorin. Regulaattoreissa on myös sisäinen virtasuojaja. Regulaattorin tehtävä

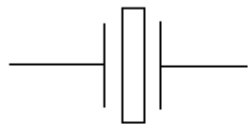
on pitää lähtöjännite tasaisena kuormasta riippumatta. Regulaattorin navat ovat in, gnd ja out. Työssä on käytetty 7805-regulaattoria, jonka numerosarja 78 ilmaisee kyseessä olevan positiivisen jännitteen regulaattori ja 05 5 voltin ulostulojännitettä.



Kuva 9. Regulaattorin piirrosmerkki

4.2.3 Kiteet

Kide on komponentti, joka värähtelee ominaistajuudellaan. Hyvyysluvusta johtuen kide on erittäin vakaa värähtelijä. Vikakoodinlukijassa kidettä on käytetty kontrollerin kellopulssin muodostamiseen.



Kuva 10. Kiteen piirrosmerkki

4.3 Muut

Tämän otsikon alla olevat komponentit eivät lukeudu passiivisiin eikä aktiivisiin komponentteihin, joten ne esitellään erikseen.

4.3.1 LCD

Vikakoodinlukijassa käytetty näyttö on 4x20 merkinen LCD LED-taustavalolla. Näytön protokollana toimii Hitachin HD47780-standardi. Näyttö on kytketty mikrokontrolleriin 4-bittisen dataväylän välityksellä.

Taulukko 3. LCD:n pinout

Pin No.	Symbol	Level	Description
1	V _{SS}	0V	Ground
2	V _{DD}	5.0V	Supply Voltage for logic
3	V _O	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→Module) L: Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	Data bus line
9	DB2	H/L	Data bus line
10	DB3	H/L	Data bus line
11	DB4	H/L	Data bus line
12	DB5	H/L	Data bus line
13	DB6	H/L	Data bus line
14	DB7	H/L	Data bus line
15	A	—	LED +
16	K	—	LED -

4.3.2 Kontrolleri

Työssä on käytetty Microchipin 33:n I/O:n omaavaa PIC 16F877A- mikrokontrolleria. Kyseisessä kontrollerissa suurin kellotaajuus on 20MHz, siinä on 8Kt ohjelmamuistia ja sen työssä käytetty versio on DIP40-koteloinen. (MicroChip)

5 LAITTEET JA OHJELMAT

Tässä insinööriyössä käytettiin elektroniikan suunnitteluun National Instrumentsin valmistamia Multisim ja Ultiboard-ohjelmia. Multisim-ohjelmassa on myös erinomainen simulaatio mutta tällä kertaa en sitä hyödyntänyt johtuen siitä, että kontrollerin simuloiminen kytkennässä vaatii C-kielisen ohjelmakoodin lisäämisen ohjelmaan. Releet piti myös luoda itse, joten niiden simuloiminen on mahdollista. Mikrokontrollerin ohjelman suunnitteluun käytin graafista Flowcode-ohjelmistoa. Nämä ohjelmat ovat tulleet tutuksi jo aikaisemmin niihin liittyvissä kursseissa, joten niiden sisäistämiseen ei mennyt aikaa vaan pääsi suoraan tekemään vikakoodinlukijan kytkentöjä ja ohjelmaa.

Piirilevyn valmistamiseksi piti tutustua myös uusiin ohjelmiin ja laitteisiin. Nämä ovat CircuitCAM piirilevyn muokkaukseen ja BoardMaster kaivertimen ohjaamiseen sekä itse kaiverrin ProtoMat c30/s.

5.1 Matrix Flowcode

Matrixin valmistama Flowcode on graafinen ohjelmointiympäristö, joka mahdollistaa kontrollereiden ohjelmoinnin ilman merkittävää koodaustaitoa. Merkittävää siksi, että kaikkia kontrollereiden toimintoja ja ominaisuuksia ei Flowcodestakaan löydy, joten ohjelmoija joutuu turvautumaan C-kieleen täydentääkseen ohjelmaa. Flowcode on graafisuutensa vuoksi selkeä ja helppokäyttöinen. Myös helppokäyttöinen, hidastettava simulaatio ja muuttujien arvojen näkymien helpottavat ohjelman tekoa. Flowcode tukee PIC-kontrollereiden lisäksi dsPIC-, PIC24-, AVR- ja ARM – kontrollereita.

5.2 National Instruments Multisim

Multisim on sähköisten kytkentöjen suunnitteluun, piirtämiseen ja simulointiin tarkoitettu ohjelma. Multisim-ohjelmasta löytyy kattava valikoima eri komponentteja, niin reaali- kuin virtuaalikomponentteja. Reaalikomponenteilla on olemassa oleva footprint, virtuaalikomponenteilla taas pelkkä simulointimahdollisuus. Vaikkei kaikkia haluttuja komponentteja löydykään, on komponentin luominen Component wizard-toiminnolla suhteellisen helppoa. Tietokannasta löytyy myös laaja valikoima valmiita footprintteja, joita voi yhdistää luomiinsa komponentteihin. Tässä työssä oli käytössä Multisim 10.1-versio.

5.3 National Instruments Ultiboard

Ultiboard on piirilevyjen suunnitteluun tehty ohjelma, jossa pääsääntöisesti hyödynnetään Multisimillä tehtyä kytkentää ja komponenttien määrittelyä. Piirilevyn voi suunnitella Ultiboard-ohjelmalla myös ilman Multisimillä tehtyä kytkentää, mutta se on mielestäni hankalaa ja siksi ilman erityistä syytä typerää. Ultiboardissa on käteviä toimintoja kuten Autoroute ja 3D-mallinnus. Autoroute-toiminto tekee johdinvedot komponenttien asetteluun mukaan välttämättä hyppylankoja. Mikäli johdotus ei autoroutella onnistu, on vielä syytä tarkastella komponenttien sijoittelua uudestaan. 3D-mallinnusta voi hyödyntää footprinttien osalta, eli tarkastella komponenttien kotelotyyppien oikeellisuutta.

5.4 CircuitCam 4.0

CircuitCAM on tietokoneavusteinen (CAD) suunnittelu- ja valmistusohjelma (CAM). Ohjelmalla käsitellään piirilevyn ominaisuuksia, kuten piirilevyn johtimien välisten eristyskanavien kokoa sekä määritellään piirilevyn läpilyöntien kohdat. CircuitCAM-ohjelma on suunniteltu nimenomaan LPFK-yhtiön valmistamille tietokonepohjaisille kaiverruslaitteille.

5.5 BoardMaster 4.0

BoardMaster- ohjelma on CircuitCAM:n tavoin suunniteltu vain LPKF-yhtiön kaivertimille eikä siinä ole toimintoja piirilevyn muokkaukseen. Ohjelma suorittaa ProtoMa –kaivertimen ohjauksen ja neuvoo käyttäjää vaihtamaan teriä kaivertimen silloisen tarpeen mukaan. Ohjelmalla määritellään piirilevyn koko ja sijoituksen kohta piirilevyaihiolle.

5.6 Protomat

Protomat on LPKF-yhtiön valmistama piirilevyn kaiverrin. Kaiverrinta ohjataan BoardMaster-ohjelmalla PC:n kautta. Kytkeä tietokoneeseen muodostetaan RS232-väylällä.

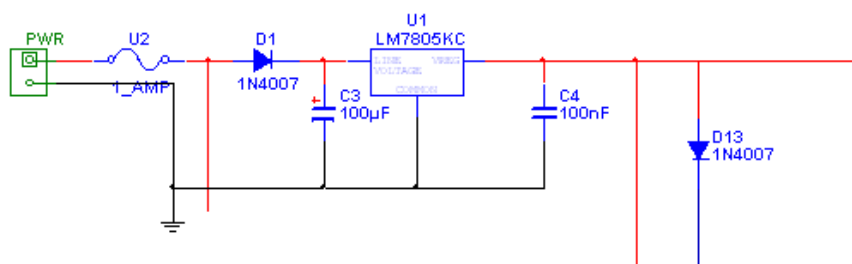
6 KYTKENTÄKAAVIO

Vikakoodinlukijan kytkentäkaavio on tehty Multisimilla ja on liitteessä 1.

6.1 Piirikaavion analysointi

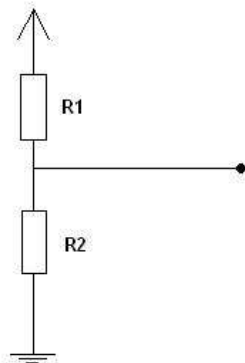
Käyttöjännite tuodaan laitteeseen PWR:ksi (power) nimetystä liittimestä.

Seuraavaksi on sulake U2 ja sen virrankestoisuus on yksi ampeeri. Sulakkeen tehtävä on katkaista virta laitteesta toimintahäiriön (komponentti oikosulussa) seurauksena ja näin suojata sekä autoa että vikakoodinlukijaa suuremmilta vaurioilta. Diodi D1 estää käyttöjännitteen väärin päin kytkemisestä aiheutuvat suuremmat tuhot kuten elkon C3 räjähtämisen tai mikrokontrollerin tuhoutumisen. Diodi myös suodattaa auton generaattorista tulevat vaihtotaajuiset häiriöt. Laitteen käyttöjännitteen tasaisuudesta ja 5 voltin linjasta huolehtii 7805-regulaattori U1. Elko C3 ja polko C4 estävät regulaattorin haitallisen värähtelyn. Kytkeä luultavasti toimisi ilmeisesti näitä kondensaattoreita, sillä värähtely ei ole yleistä. Kondensaattorien arvot ovat yleisesti käytettyjä, tosin datasheet antaa C3:n arvoksi 0,33uF ja C4:n arvoksi 0,100uF(lähde). Diodi D13 on kytketty regulaattorin lähdön ja mikrokontrollerin käyttöjännitteen väliin. Tämän kytkennän tehtävänä on estää ICSP-väylän kautta ohjelmoimassa virran kulku muihin komponentteihin kuin itse kontrolleriin. Ilman diodia virran kulutus olisi huomattavasti suurempi LCD:n taustavalon takia ja ICSP:n jännite notkahtaisi, eikä kontrollerin ohjelmointi onnistuisi. Kuvassa 11 nähdään laitteen käyttöjännitteen suodatus.



Kuva 11. Käyttöjännitteen suodatus

Vastukset R11 ja R12, R17 ja R18 sekä R19 ja R20 muodostavat jännitteenjakokytkennän. Tämä kytkentä pudottaa auton järjestelmien noin 0-14 voltin jännitteet controllerille sopiviksi 0-5 voltin jännitteiksi. Kuvassa 12 havainnollistetaan jännitteenjakokytkentä vastuksilla.



Kuva 12. Jännitteenjakokytkentä

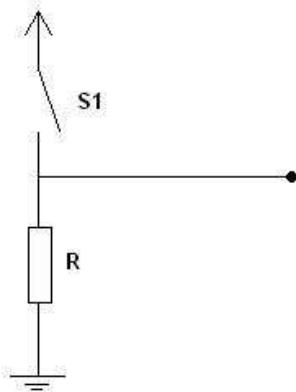
Jännitteenjakokytkennällä saadaan stabiili jännite kuormaan, tässä tapauksessa controllerin tuloon. Kytkennän ulostulojännite saadaan kaavasta $U_{out}=U_{in} \times R2/(R1+R2)$. Kuten aikaisemmin todettiin, controllerin tulo ei niele virtaa johtuen suuri-impedanssisuudesta, ei kaavassa huomiotta jätetty kuorman vaikutus jännitteisiin vaikuta lopputulokseen.

Näissä kytkennöissä R1 arvo on 10k Ω ja R2 on 5,6k Ω .

Tulosignaalin ollessa 14,4V saadaan kytkennästä $U_{out} = 14,4V \times 5,6k\Omega/(10k\Omega + 5,6k\Omega) = 5,2V$. Kytkentää voidaan pitää hiukan alimitoitettuna, koska se ylittää 5 voltin ihanteen. Osa vikakoodeista luetaan kuitenkin auton ollessa sammuksissa, jolloin signaalit vaihtelevat 0-12,7 voltin amplitudilla. Tässä tilanteessa jännitteenjaon ulostulojännite on 4,55 voltia. Tämä on vielä riittävä, jotta controlleri tulkitseisi jännitteen loogiseksi tilaksi 1. Kytkentä toimii siis hyvin, sillä controlleri sietää tässä käsitellyt jännitteet(Microhip lähde).

R10 sekä R21-R25 ovat controllerin tuloihin (A1-A5 ja E0) liitettyjen painonappien alasetovastuksia. Ilman näitä vastuksia olisivat tulot kelluvia ja vikakoodinlukijan toiminta sekavaa. Painonapit puuttuvat liitteen 1 kytkentäkaaviosta, sillä ne ovat johdotettu erilleen piirilevystä. Painonappien tilalla on 7-napainen S:ksi (Switch) nimetty liitin. Kuusi liitäntää kytketään painonappien toiseen liitäntään ja seitsemäs 5 voltin käyttöjännitteeseen.

Kuvassa 13 on painonapin kytkentä mikrokontrolleriin, jossa S1 on painonappi ja R vastus.



Kuva 13. Painonapin kytkentä kontrolleriin

Kontrolleri vaatii tuloon joko loogisen nollan tai ykkösen, TTL- tason jännitteinä 0 voltia tai 5 voltia. Koska kontrollerin tuloimpedanssi on suuri, ei I/O pinniin kulje virtaa eikä jännitehäviötä synny vastuksen R yli, on kytkimen ollessa auki kytkennän lähdössä 0V. Kytkimen S1 ollessa kiinni, on lähdössä 5 voltia. Vastuksen R tehtävä on siis estää oikosulku kytkimen ollessa kiinni ja toisaalta vetää kytkennän lähtö alas kytkimen ollessa auki. Työssä vastuksen R arvo on 10k Ω ja tehonkesto 0,25W. Kaavasta $U=RI$, jossa U jännite, R resistanssi ja I virta, saadaan kytkimen ollessa kiinni vastuksen jännitehäviöksi 5V ja virraksi 0,5mA. Vastuksessa tapahtuva tehohäviö saadaan tehon kaavasta $P=UI$ eli $P=5V \times 0,5mA = 0,0025W$. Tehohäviö on siis pienempi kuin vastuksen tehonkesto ja näin ollen vastus kestää kytkennässä.

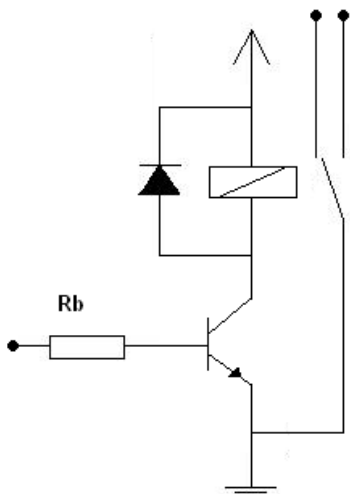
Vastukset R5 ja R7 sekä 100 Ω trimmeri muodostavat myös jännitteenjakokytkennän, jonka ulostulo viedään ADC0-kanavaan. Trimmerillä saadaan ulostulojännitettä hienosäädettyä, jolloin A/D-muuntimella saadaan jännite tarkasti mitattua. Laitteen ensimmäisessä versiossa ei ollut diodia D13, jolloin kontrollerin käyttöjännite oli suurempi. Tällöin ADC:n jännitteenjaossa oli myös vastus R6. Uuden mitoituksen takia R6 voitiin jättää pois, mutta se näkyy osalistassa ja piirilevyn suunnittelussa. Jännite kytkentään on otettu ennen diodia D1, sillä diodin kynnyksjännite riippuu lämpötilasta suhteessa 1,8mV/1 $^{\circ}$ C (Timo Haiko). Vaikka vaikutus ei ole suuri eikä jännitemittauksen tarpeen näyttää millivoltin tai kymmenenkään millivoltin tarkkuudella, on kynnyksjännitteen riippuvuus lämpötilasta hyvä ottaa huomioon ainakin yleisellä tasolla.

Vastukset R1, R8 ja R9 vetävät ICSP-väylän alas, jottei laite menisi ohjelmointitilaan tarkoituksettomasti. Myös kaikki käyttämättömät I/O -pinnit on vedetty alas toimintahäiriöiden välttämiseksi.

100 Ω vastus R4 vetää MCLR:n ylös, sillä tämän kintun alasetäminen aiheuttaa ohjelman suorittamisen resetoinnin. Ilman tätä vastusta laitteen toiminta on myös epämääräistä. Vastus R2 rajoittaa

näytön LED -taustavalon kuluttamaa virtaa. Vastus R3 asettaa LCD:n kontrastin. Kontrasti on suurimmallaan nollassa voltissa ja pienimmillään 5 voltin linjassa.

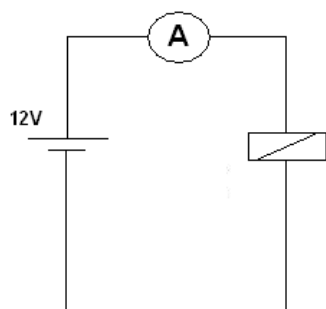
Kahdeksannapaisen, pulssimuotoisen väylän lukemisen aloittamiseen ja nollaukseen on käytetty releitä. Releet sulkevat väylän jännitteen nollatasoon painonapin tavoin. Releen käämin vaatimaa 30 milliampeerin ohjausvirtaa kontrolleri ei kuitenkaan kestä (MicroChip), joten työssä on käytetty transistoria kytkimenä ohjaamaan relettä.



Kuva 14. Transistori ohjaamassa relettä

Kun kantavirta I_b ei kulje, on transistori avoimessa, korkeaimpedanssisessa tilassa eikä releen käämi saa näin nollopotentiaalia eikä virtaa kulje käämin läpi. Kun kantavirta I_b kulkee, transistori avautuu ja virta kulkee releen käämin läpi transistorin kollektorin kautta emitterin läpi maihin. Virtapiiri siis sulkeutuu ja rele vaihtaa tilaa.

Tarvittavan kantavastuksen R_b koon selvittämiseksi täytyy tietää releen vaatima virta. Mittasin virran kytkemällä releen jännitelähteen ja virtamittarin väliin. Releen kelan nimellisjännite on 12 voltia, jonka säädin säädettävästä jännitelähteestä kohdalleen. Mittaustulokseksi sain noin 30 milliampeeria, joten kollektorivirran suuruus I_c on siis 30mA. Transistorin minimi tasavirtavahvistuskertoimeksi valmista antaa arvon 75 (STMicroelectronics). Kantavirta I_b saadaan kaavalla $I_b = I_c / h_{FE} = 0,03A / 75 = 0,4mA$. Koska saatu virta on pieni, jolla saadaan vahvistukseltaan huonoimman transistorin kollektorivirta halutun suuruiseksi, on hyvä suurentaa arvoa hiukan. Tällä varmistetaan transistorin meneminen kyllästystilaan, sillä aktiivitilassa osa tehosta hukkuisi transistoriin muodostaen lämpöä. Insinööriyössä arvoksi on päätetty 2mA.



Kuva 15. Virran mittaus releen kelasta

Kantavastuksen R_b koko saadaan kaavasta $R_b = (U - U_{BE}) / I_c = (5V - 0,7V) / 0,002A = 2150 \Omega$. Pyöristetään saatu arvo lähimpään saatavilla olevaan eli $2,2k \Omega$.

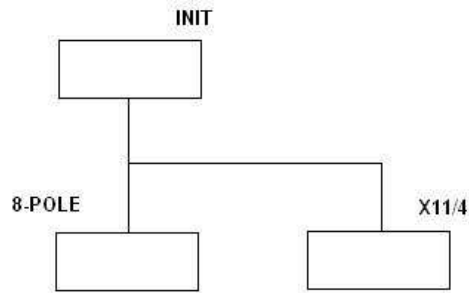
Diodin tehtävä kytkennässä on estää transistorin tuhoutuminen kelan läpi kulkevan virran katkaisuhetkellä. Kela vastustaa siinä tapahtuvia virranmuutoksia ja näin ollen virran katkaisuhetkellä siinä muodostuu jännitteeltään vastakkainen, jopa satojen volttien suuruinen induktiosysäys.

Kun mitoitus on tehty, on syytä verrata saatuja arvoja transistorin maksimiarvoihin (lähde) ja voimme todeta transistorikytkennän mitoituksesta saadut arvot oikeellisiksi.

Kerkot C1 ja C2 ovat varmistamassa kiteen oikeanlaisen värähtelyn.

7 OHJELMA

Ohjelma on tehty Matrix Flowcode V4:llä. Tässä kappaleessa käsitellään vain ohjelman tärkeimmät kohdat vuokaavion laajuuden takia. Laitteen käyttöliittymä on tehty englanniksi, koska vikakoodit ja työssä käytetyt ohjelmistotkin ovat englanniksi. Johtuen kahdesta erilaisesta luettavasta väylästä, on laitteessa myös kaksi pääohjelmaa. Pääohjelmilla on yhteinen initialisointi (alkuarvojen asetus), jossa kerrotaan controllerille mihin näyttö on kytketty ja on annettu muuttujille arvot 0. Näin varmistetaan ohjelman resetoituminen täydellisesti, mikäli virransyötössä ilmenee ongelmia. Initialisoinnissa tarkistetaan onko D1- tuloon kytketyn jännitejakokytkennän lähtö looginen 1 vai 0. Mikäli tila on 1, on johdot kytketty X11/4 liittimeen jolloin mennään pulssisuhdetta ja livedatan mittausta varten tehtyyn pääohjelmaan. Myös laitteen käyttöopas on sisälletty INIT-lohkoon. Pääohjelmat ovat nimetty luettavien väylien liittimien mukaan. Pääohjelmat ovat omien looppiensa sisällä, joiden ehtona on while 1. Tämä tarkoittaa ohjelman pyörivän loopissa niin kauan kuin virrat on kytketty. Kuvassa 16 on ohjelman päälohkot.

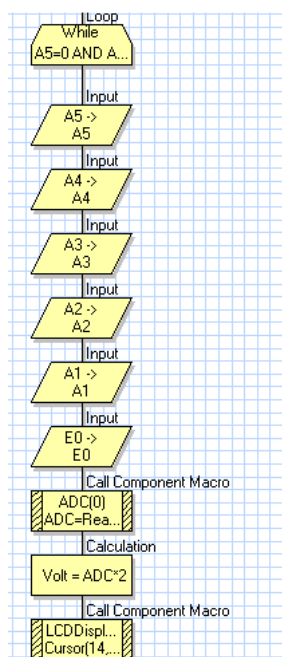


Kuva 16. Ohjelman päälohkot

7.1 Pulssimuotoinen väylä

Tutustutaan ensin 8-napaisen liittimen pulssimuotoiseen väylän lukemiseen. Pääohjelman alussa, ennen looppia, enableidaan timer0 keskeyttämään 75Hz:n taajuudella. Myös portB keskeytys sallitaan. Jokaisella keskeytyksellä lisätään integer -tyyppiseen, tmr0 -nimiseen muuttujaan arvo yksi kaavalla $tmr0 = tmr0 + 1$. Tätä muuttuja hyödynnetään päätettäessä onko saapuvia pulsseja ja saapuvien pulssien loppumisesta.

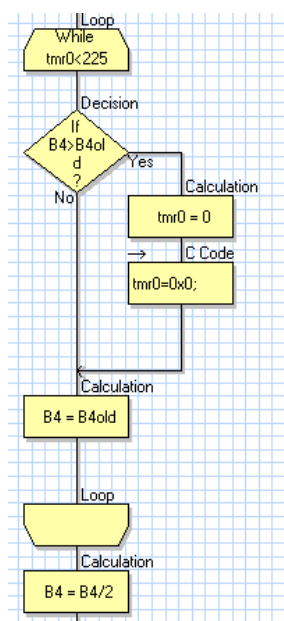
Näyttöön kirjoitetaan LCD-komponenttimakrolla mahdolliset lisälaitteiden nimet ja niitä vastaavat painonapit. Tämän jälkeen ohjelma pyörii loopissa lukemassa painonappien tilaa kunnes joku niistä on yksi. Loopissa on myös jännitteen mittaus A/D-muuntimen kautta ja tulostus näytölle. ADC:n lukeminen tapahtuu komponenttimakrolla ADC read, jossa arvo tallennetaan ADC-nimiseen int-muuttujaan. Saatu arvo kerrotaan kahdella saadaksemme todellisen jännitteen ja arvo tallennetaan Volt -muuttujaan ja tulostetaan näytölle.



Kuva 17. Kuvakaappaus painikkeiden ja A/D muuntimen lukemisesta

Loopin jälkeen on if-ehtoja, joissa tarkistetaan painettu nappi ja asetetaan sitä vastaava lähtö arvoon 1 sekunniksi. Näin lähtöön kytketty rele ohjaa kyseisen väylän maihin sekunniksi ja vikakoodin luku alkaa.

Pulssit tallennetaan tulon mukaan nimettyyn B4-muuttujaan ja entinen pulssien lukumäärä tallennetaan B4old-muuttujaan. Mikäli uusia pulsseja saapuu, timer0 resetoitetaan C-kielisellä koodilla `tmr0=0x0`; ja kolmen sekunnin looppi alkaa alusta. Muutoin aika täyttyy ja ohjelma putoaa loopista pois. Koska portB-keskeyttää sekä nousevalla että laskevalla pulssin reunalla (MicroChip), täytyy saatu pulssien lukumäärä jakaa kahdella.



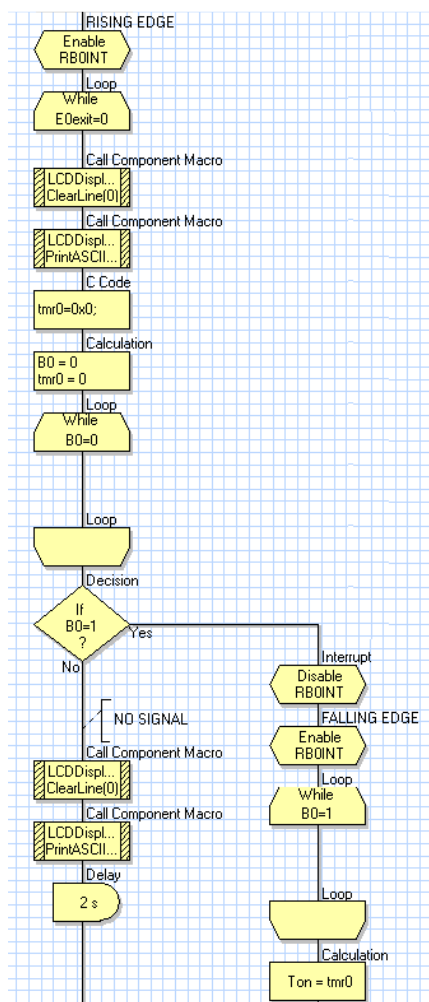
Kuva 18. Kuvakaappaus pulssien lukemisesta

Ohjelman laskettua pulssien lukumäärät tarkistetaan if-ehdoilla, onko pulssien lukumäärä 0 tai 1. Pulssien ollessa nolla ei kyseistä lisävarustetta ole kytketty tai siinä on häiriö. Ohjelma kirjoittaa näytölle tekstin no signal. Pulssien ollessa 1 ei laitteessa ole vikaa ja näyttöön tulostetaan teksti no fault found. Näissä tapauksissa koodin tyhjennysvaihtoehtoa ei tulosteta näytölle. Mikäli pulssi on lukumäärältään jotain muuta, mahdollistetaan if-ehdoilla pulssien tyhjennys ja exit. Mikäli clear-toimintoa vastaavaa nappia painetaan, ohjelma nostaa kyseisen lähdön ylös kahdeksaksi sekunniksi, jolloin koodi tyhjentyy muistista. Exit vie takaisin päävalikkoon, josta voidaan valita halutun laitteen lukeminen.

7.2 Pulssisuhde ja live-data

Ennen tämän pääohjelman looppia enableidaan timer0 ja asetetaan keskeytystaajuudeksi 19,2kHz. Ohjelma pyytää valitsemaan lIVEDATAN tai pulssisuhde-mittauksen ja looppaa painonappeja kunnes toinen on valittu. Loopista siirrytään if-ehtojen perusteella oikeaan kohtaan ja aletaan suorittaa mittauksia. Mittauksesta pääsee jälleen takaisin painikkeella E0, josta ilmoitetaan näytössä.

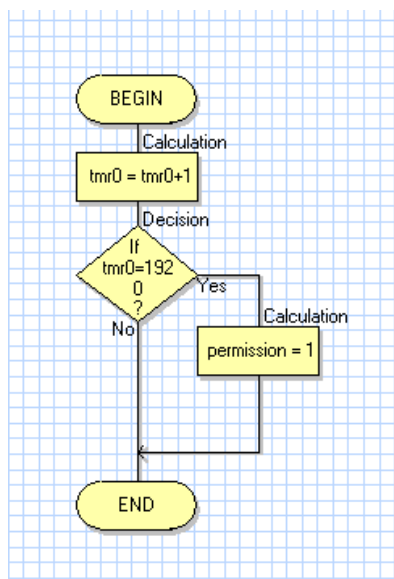
Pulssisuhde mitataan sallimalla ensin B0int-keskeytys pulssin nousevalla reunalla ja loopataan kunnes tämä tapahtuu. Mikäli pulssia ei tule 100 millisekunnin aikana, ohjelma ilmoittaa näytöllä no signal ja aloittaa pulssisuhteen mittaamisen alusta. Tämän jälkeen B0int disabloidaan ja enableidaan laskevalle reunalle. Mikäli laskevaa reunaa ei tule, ilmoittaa laite signaalin olevan aina ylhäällä tekstillä signal always up. Laskevan reunan saapuessa tallennetaan timer0:n arvo muuttujaan Ton, disabloidaan laskeva reuna ja enableidaan jälleen nouseva. Nousevan reunan aiheuttaessa keskeytys tallennetaan timer0:n arvo muuttujaan Tfull. Näin on tiedossa signaalin ylhäällä olo -aika ja jaksonaika jolloin pulssisuhde saadaan kaavalla $Duty = \frac{Ton}{Tfull}$. On kuitenkin huomattava Boschin järjestelmässä käytetty yleisestä poikkeava pulssisuhde, jossa verrataan alhaalla olo-aikaa ylhäällä olo-aikaan. Täytyy myös huomioida int-muuttujan olevan kokonaisluvulle joten kerrotaan Ton arvo sadalla. Näin ollen ohjelmassa käytetään kaavaa $DUTY = \frac{100 - Ton * 100}{Tfull}$. Lukuisista yrityksistä huolimatta pulssisuhteen mittausta ei toimi luotettavasti. Joko sitä ei lasketa ollenkaan tai siinä on toleranssia.



Kuva 19. Kuvakaappaus pulssisuhteen mittauksesta

Live-datan arvoja vikakoodinlukijassa edustaa jännite, kierrosluku ja sytytysennakko. Sytytysennakko oli alun alkaenkin ekstra ohjelmassa, joten ajan puutteen takia tämä jäi toteuttamatta. Kierrosluku saadaan laskemalla pulssien lukumäärä B4-keskeytyksen kautta 100 millisekunnin aikana. Timer0-makrossa lisätään jokaisella keskeytyksellä tmr0-muuttujaan arvo 1. Saavuttaessaan arvon

1920 on siis 100ms kulunut ja if-ehdolla annetaan lupa laskea ja tulostaa kierrosluku näytölle. Kierrosluvun arvo päivittyy siis 10 kertaa sekunnissa.



Kuva 20. Kuvakaappaus Timer0-makrosta

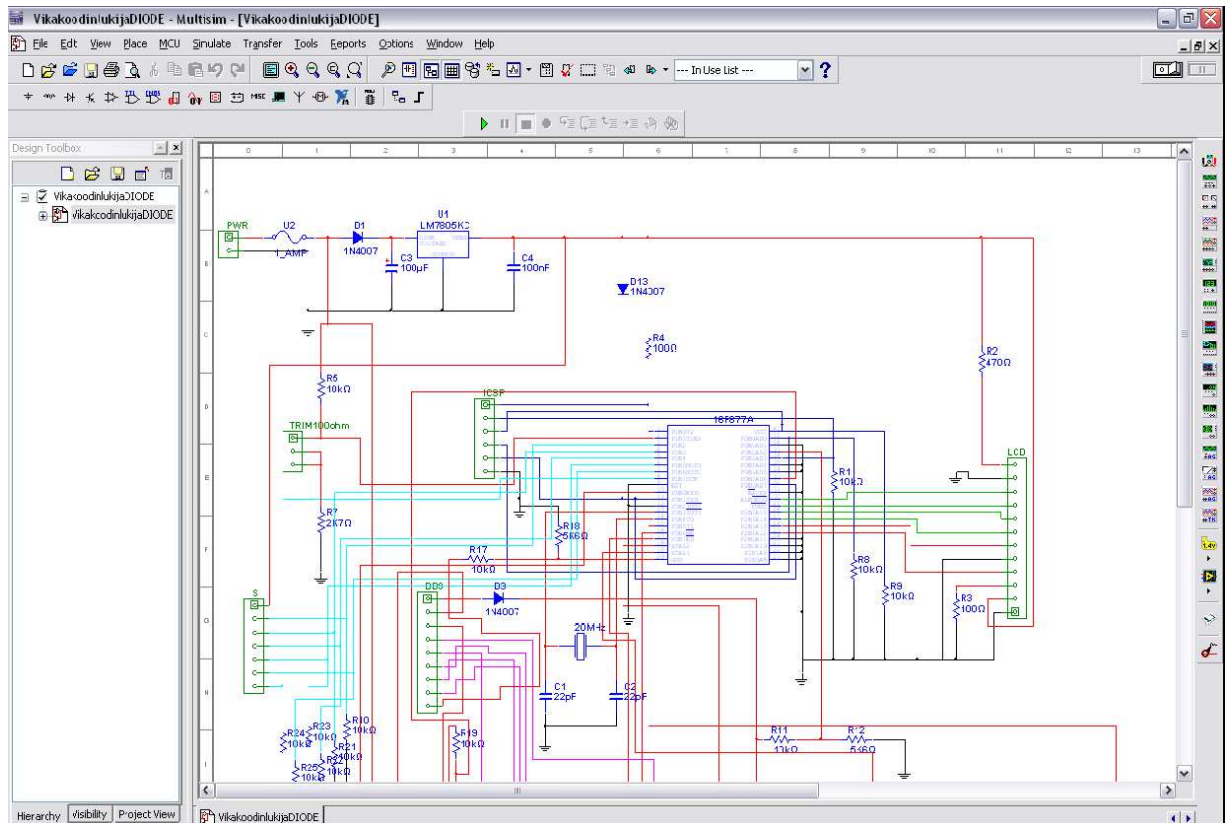
Yhdellä moottorin kierroksella väylästä tulee 4 pulssia ja jokaisesta pulssista kaksi keskeytystä. Kierrokset ilmoitetaan kierrosten lukumäärällä minuutissa. Näin ollen kierrosluku saadaan ohjelmassa kaavalla $RPM = B4 * 5 * 30$.

8 TYÖN TOTEUTUS

Tässä kappaleessa kerrotaan itse työn toteutuksesta ja se käydään vaihe vaiheelta läpi.

8.1 Kytkennän piirtäminen

Työ alkoi laitteen elektroniikan suunnittelulla Multisimillä hyödyntäen. Multisimin ylävalikosta löytyvät komponentit ja muut työkalut. Mikäli jotain komponenttia ei löydy, voi sen myös luoda itse component wizard-toiminnolla. En löytänyt ohjelmasta sopivaa relettä, joten loin uuden komponentin ja sijoitin siihen multisim tietokannasta löytyneen oikeanlaisen footprintin. Footprintit on syytä katsoa kohdilleen jo Multisimin puolella, sillä niiden vaihtaminen Ultiboardissa ei ole niin helppoa. Mikäli footprint tai kytkentä tuli kuitenkin virheellisesti valittua ja erehdys huomataan vasta Ultiboardissa piirilevyn suunnittelun olevan kesken tai valmis, voidaan käyttää forward annotate-toimintoa. Menään siis takaisin Multisimiin, korjataan footprint tai kytkentä oikeaksi, tallennetaan ja siirretään forward annotate-toiminnolla takaisin Ultiboardiin. Näin jo suunniteltu piirilevy ei mene hukkaan vaan kaikki muutokset päivittyvät jo tehtyyn työhön.



Kuva 21. Kuvakaappaus Multisimista

8.2 Piirilevyn suunnittelu

Piirilevyn suunnitteluun käytin Ultiboardia. Tiedostojen siirto Multisimista Ultiboardiin tapahtuu annon-
 te to ultiboard-toiminnolla. Tällöin ohjelma pyydetään tallentamaan ja Ultiboard avautuu. Import
 netlistiin valitaan ok ja pääsemme itse ohjelmaan.

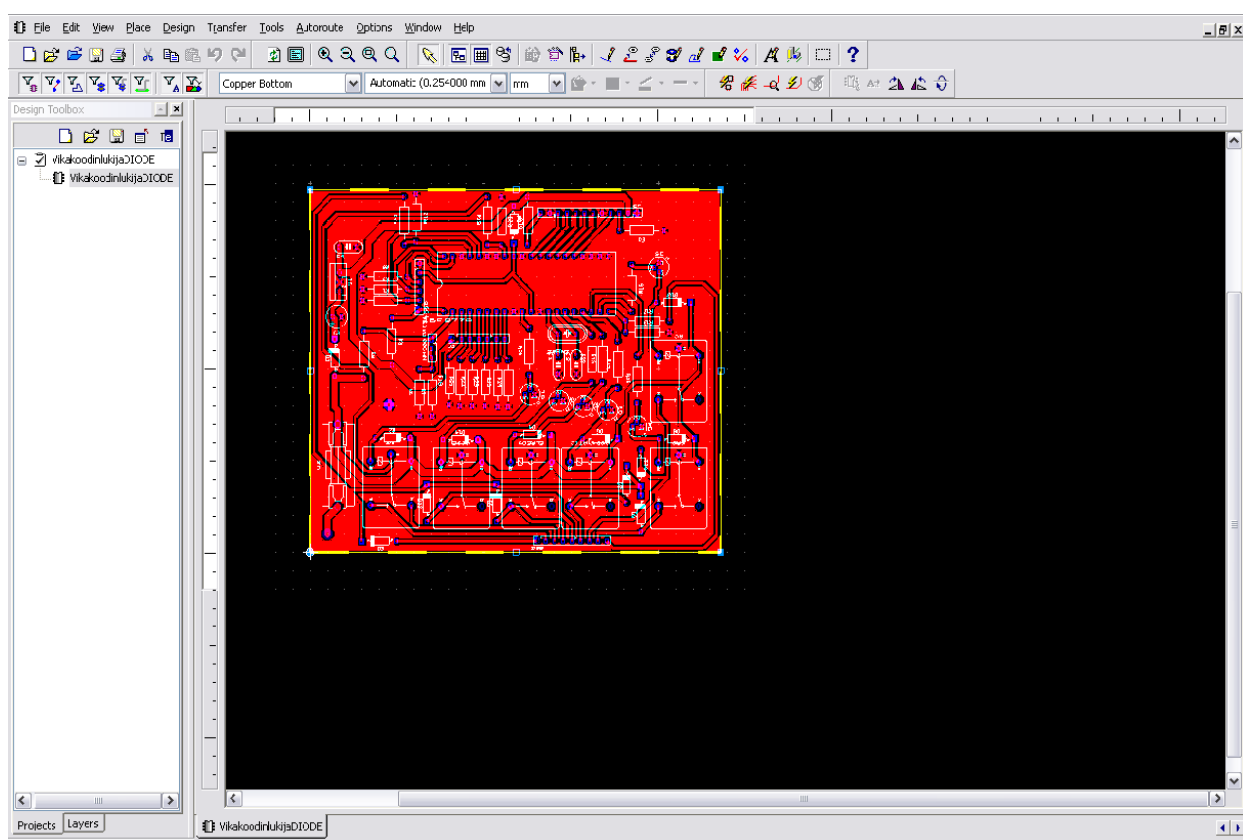
The screenshot shows the "Import Netlist" dialog box in Ultiboard. It contains a table with the following columns: Item, Action in Layout, and Status. The table lists various components and their connections to be imported from a netlist.

Item	Action in Layout	Status
Layers Copper Bottom, Copper Top	Add Layers Copper Bottom, Copper Top	
Net 0	Add 16F877A pin 9 to net 0	
Net 0	Add 16F877A pin 21 to net 0	
Net 0	Add 16F877A pin 22 to net 0	
Net 0	Add 16F877A pin 23 to net 0	
Net 0	Add 16F877A pin 24 to net 0	
Net 0	Add 16F877A pin 34 to net 0	
Net 0	Add 16F877A pin 35 to net 0	
Net 0	Add 16F877A pin 38 to net 0	
Net 0	Add 16F877A pin 12 to net 0	
Net 0	Add 16F877A pin 31 to net 0	
Net 0	Add ICSP pin 6 to net 0	
Net 0	Add R1 pin 2 to net 0	
Net 0	Add C1 pin 2 to net 0	
Net 0	Add C2 pin 2 to net 0	
Net 0	Add LCD pin 1 to net 0	
Net 0	Add LCD pin 5 to net 0	
Net 0	Add LCD pin 11 to net 0	
Net 0	Add R3 pin 2 to net 0	
Net 0	Add U1 pin 2 to net 0	
Net 0	Add C3 pin 2 to net 0	
Net 0	Add C4 pin 2 to net 0	
Net 0	Add R7 pin 2 to net 0	
Net 0	Add R8 pin 2 to net 0	
Net 0	Add R9 pin 2 to net 0	

Buttons: OK, Cancel, Help

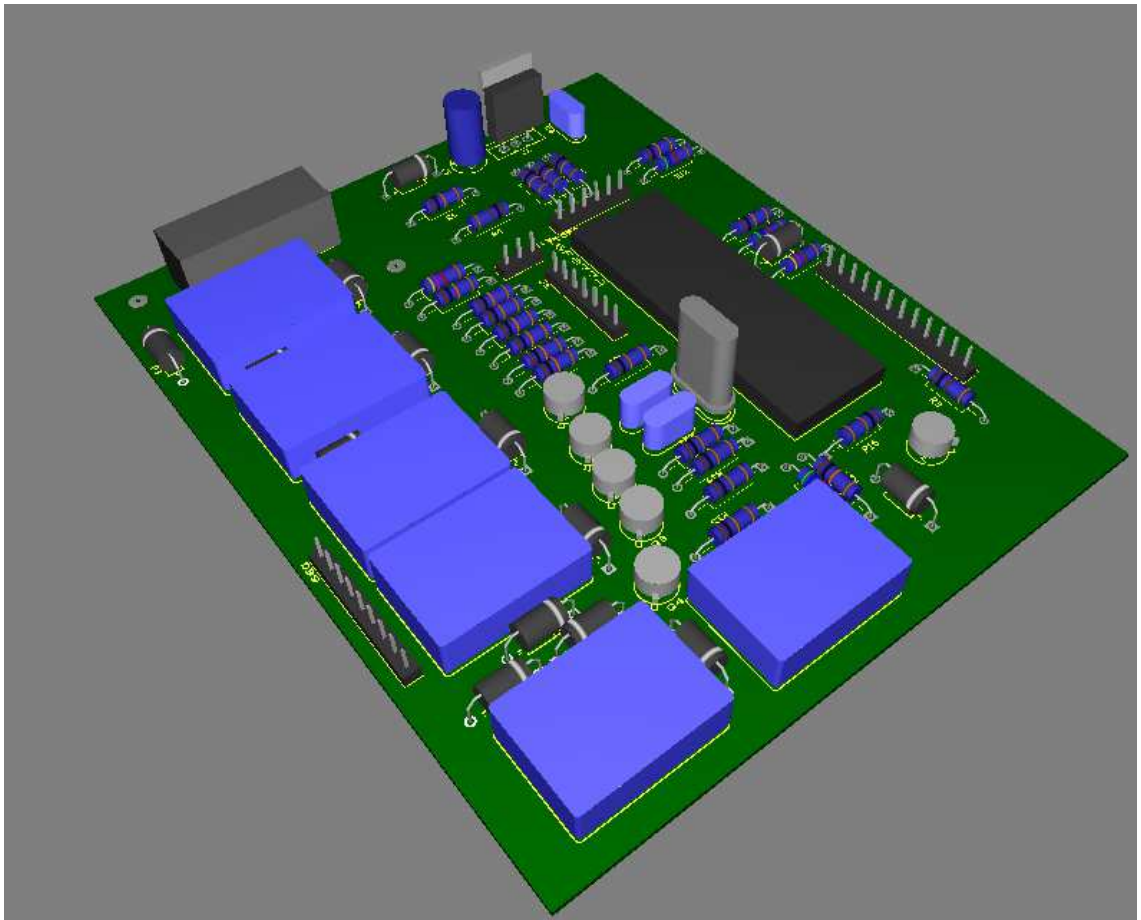
Kuva 22. Kuvakaappaus Netlististä.

Valitaan piirilevyn powerplaneeksi net 0, jolloin piirilevyn jyrshintätarve vähenee. Piirilevy muuttuu punaiseksi onnistuneesta powerplanen valinnasta. Valitaan ylhäällä olevasta työkalupalkista kaikki aktiivisiksi ja klikkaamalla hiiren oikealla piirilevyn reunoja saadaan piirilevyn kokoa muokattua. Tässä työssä piirilevyn kooksi on valittu kotelon sisämittojen perusteella 100mm x 120mm. Sijoitellaan komponentit levyille raahaamalla ja muutetaan niiden asentoa kunnes päästään mieleiseen lopputulokseen. Piirilevyn johdotukset tehdään place line- toiminnolla. Tehtyjen johtinvetojen jälkeen on syytä muokata johdinvetoja ja eristysvälejä isommiksi, sillä oletusarvoisesti ne ovat 0,254 mm. Näin ohuessa vedossa ei suuria virtoja kulje ja juottaminen olisi haastavaa. Työssä on käytetty pienimpiä arvoja molempiin 0,5 mm.



Kuva 23. Kuvakaappaus Ultiboard-ohjelmasta.

Piirilevyn suunnittelua voidaan pitää onnistuneena, mikäli siinä ei ole hyppylankoja ja levy on tasaisesti täytetty. Johdinvetojen olisi myös hyvä olla mahdollisimman lyhyitä. Suunniteltua piirilevyä on hyvä tarkastella lopuksi Ultiboard-ohjelman 3D-mallinnuksen avulla. Tästä selviää helposti väärät kotelotyypit ja mahdollinen tilanahtaus komponenttien juottamisvaiheessa. Koska kytkentä koki erilaisia muutoksia insinööriyön aikana, on piirilevy suunniteltu kuusi kertaa ennen sen valmistamista.



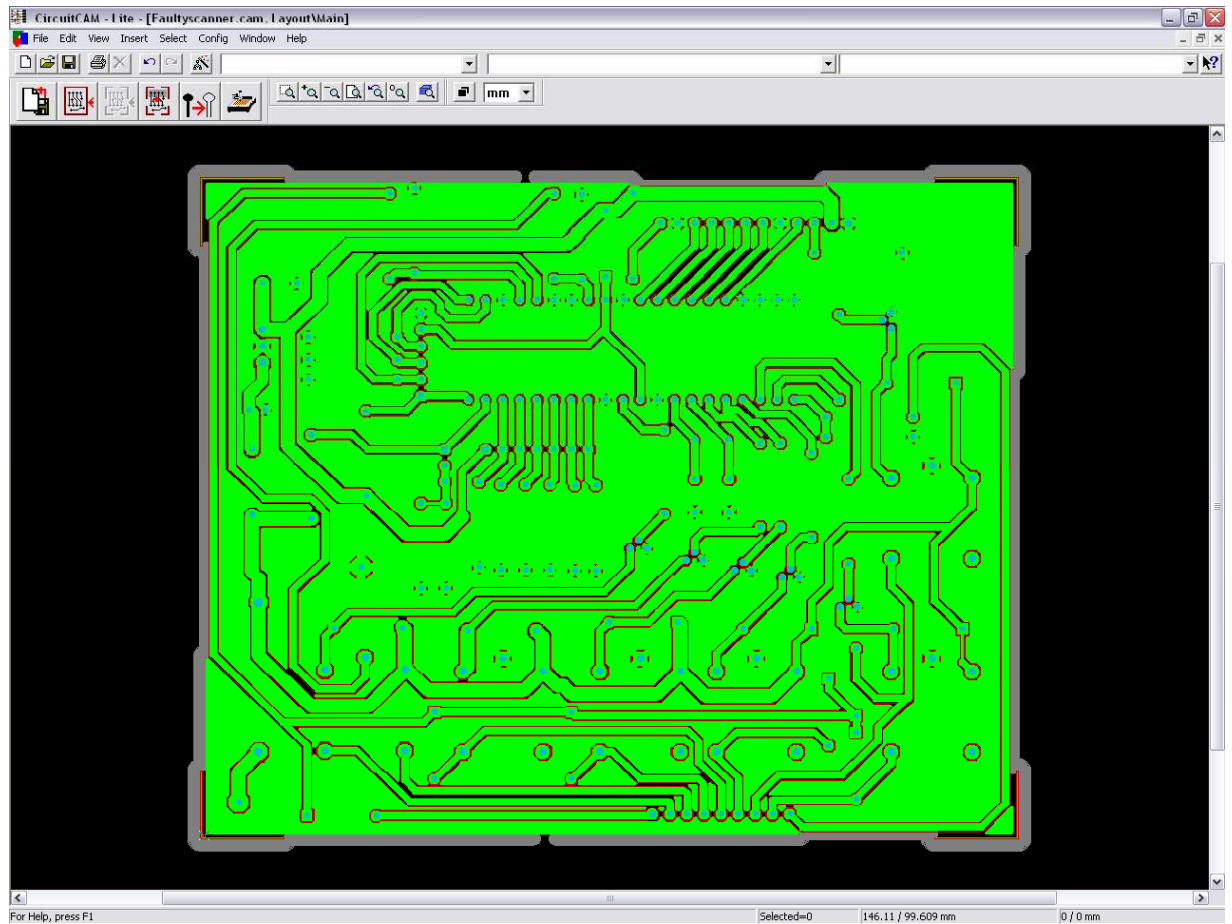
Kuva 24. Piirilevyn 3D-mallinnus

Kuvasta 24 nähdään, että transistoreilla on väärä kotelotyyppi T0-18 oikean tyyppin T0-92 sijasta. Johtuen samasta kinttujärjestyksestä ja fyysisestä koosta ei asialla ole kuitenkaan merkitystä. Myös releiden kotelot ovat matalampia kuin todellisuudessa. Lopuksi luodaan jysintätiedostot file valikosta export ja valitaan Gerber Rs-247x ja NC drill.

8.3 Piirilevyn valmistus

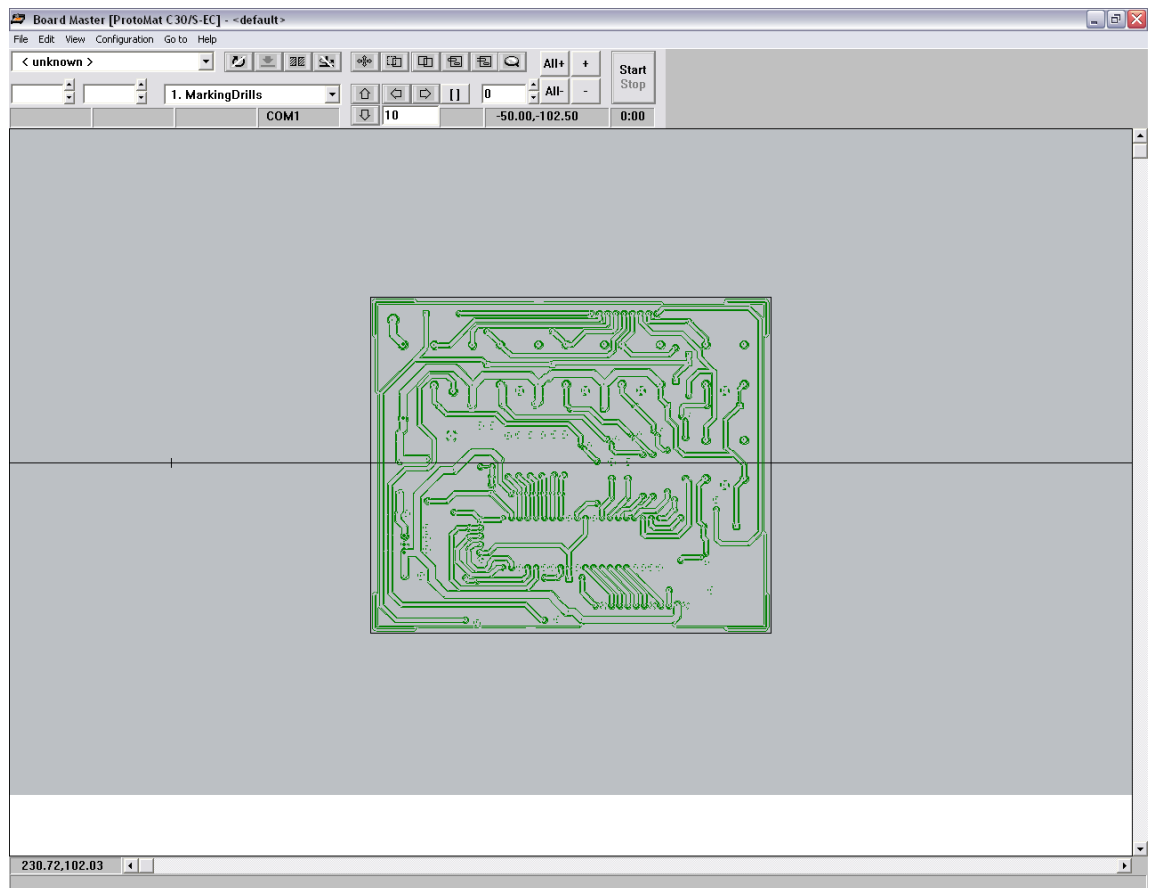
Kun piirilevy on suunniteltu, täytyy se siirtää vielä CircuitCAM-ohjelmaan, jossa määritetään piirilevyn läpilyöntirajot ja tarkastetaan johdinten välisten eristeiden koko. Tämä on viimeinen ohjelma, jossa suunniteltua piirilevyä muokataan ennen valmistusta.

Theseuksessa julkaistussa Pasi Lyyran laatimassa käyttöohjeessa (LYYRA, Pasi 2004) ohjeistetaan käyttämään Wizard-toimintoa ohjelman monimutkaisuuden vuoksi. Lyyran ohjeiden mukaisesti wizard pyytää valitsemaan tiedostoja ja asetuksia ja näiden välillä painellaan nextiä kunnes wizardissa ilmestyy finish. Koska piirilevy on suunniteltu valmiiksi hyvin, ei muokkaustarvetta piirilevylle enää ollut. Wizardissa määritetään myös piirilevyn läpilyönti. Valitsin vain kaksi kiinnityspistettä helpottaakseni piirilevyn irrottamista aihosta. Kiinnityspisteet sahasin rautasehän terällä.

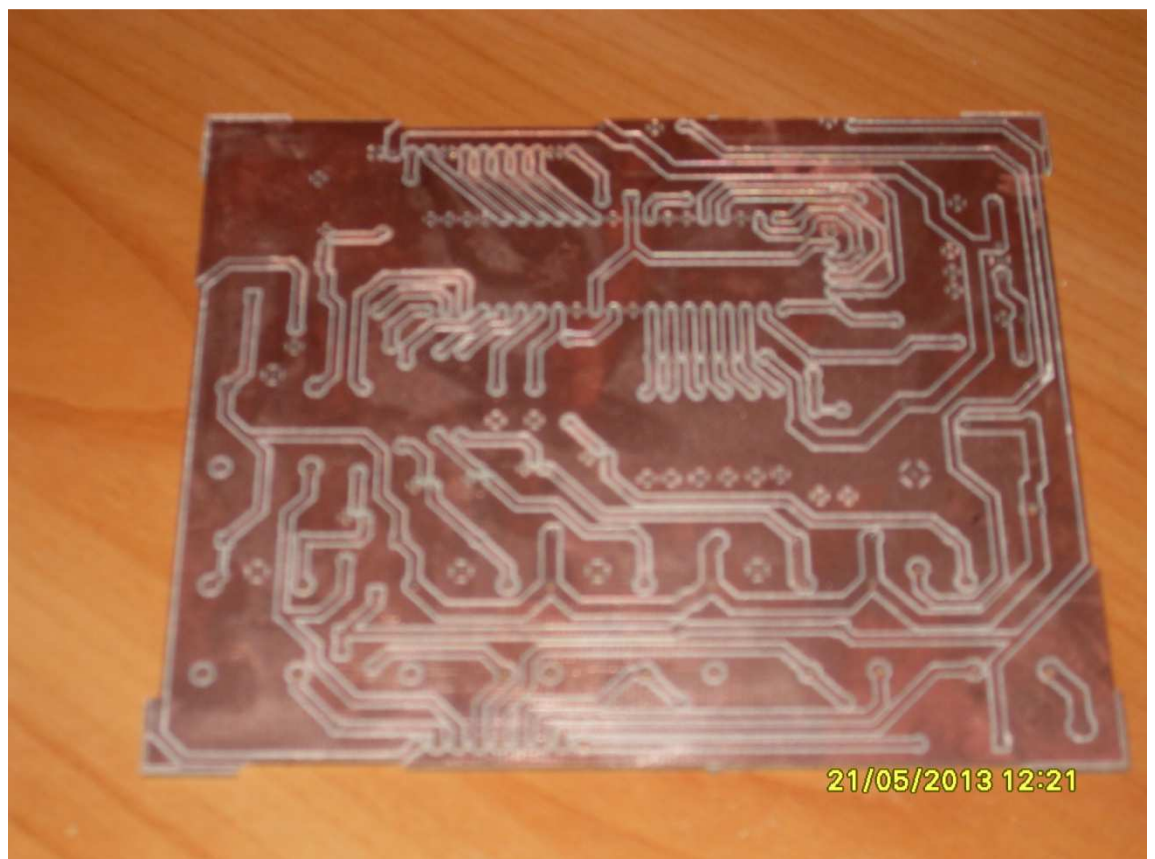


Kuva 25. Kuvakaappaus CircuitCAM-ohjelmasta.

Tämän jälkeen CircuitCAMissa luotu piirilevykuva avataan BoardMaster-ohjelmalla. Ohjelmassa määritellään piirilevyaihion koko set low corner- ja set high corner-toiminnoilla. Tämän jälkeen sijoitetaan piirilevykuva haluttuun kohtaan ja painetaan start. Jyrsin käy aina välillä vaihdattamassa haluamansa terän ja jatkaa jyrshintää. Vikakoodinlukijan piirilevyn jyrshintäessä kului aikaa hieman alle tunti.



Kuva 26. Kuvakaappaus BoardMaster-ohjelmasta.



Kuva 27. Valmis piirilevy

8.4 Juottaminen

Käytössä oli Solomonin valmistama termostaattiohjattu juotin SR-976ESD. Keraamisen juotinkärken-
sä ansiosta se on nopea lämpeämään. Juotoslämpötilana käytin 350-astetta. Komponenttien asette-
lun näkee hyvin piirilevyn 3D-mallinnuksesta. Juotettaessa on tärkeää välttää erityisesti puolijohde-
komponenttien tarpeetonta kuumentamista, sillä ne tuhoutuvat herkästi. Juottamisen jälkeen mah-
dollinen ylimääräinen pituus komponenttien jaloista katkaistaan ja tarkistetaan juotos.

Vikakoodinlukijan osaluettelo on liitteessä 2.

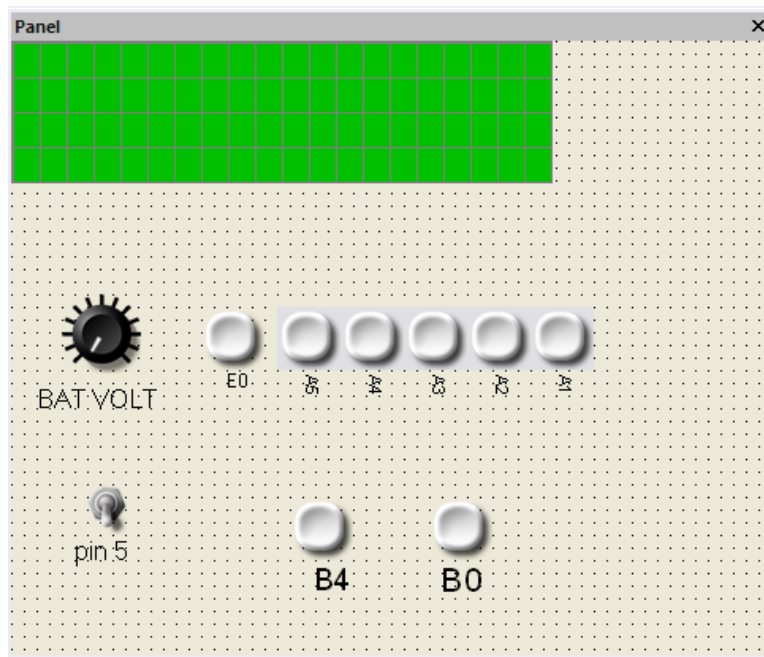
DB9 liittimeen pitää valmistaa kaksi johtoa, jossa toisessa 4mm banaaniplugit ja toisessa 2,5mm
monoplugit. Alla olevasta taulukosta selviää kytkettävät liittimet.

Taulukko 4. DB9-liittimen kytkennät

DB9	8pole	X11
1		1
2		7
3		3
4	4	
5	3	
6	5	
7	7	
8	6	
9		5

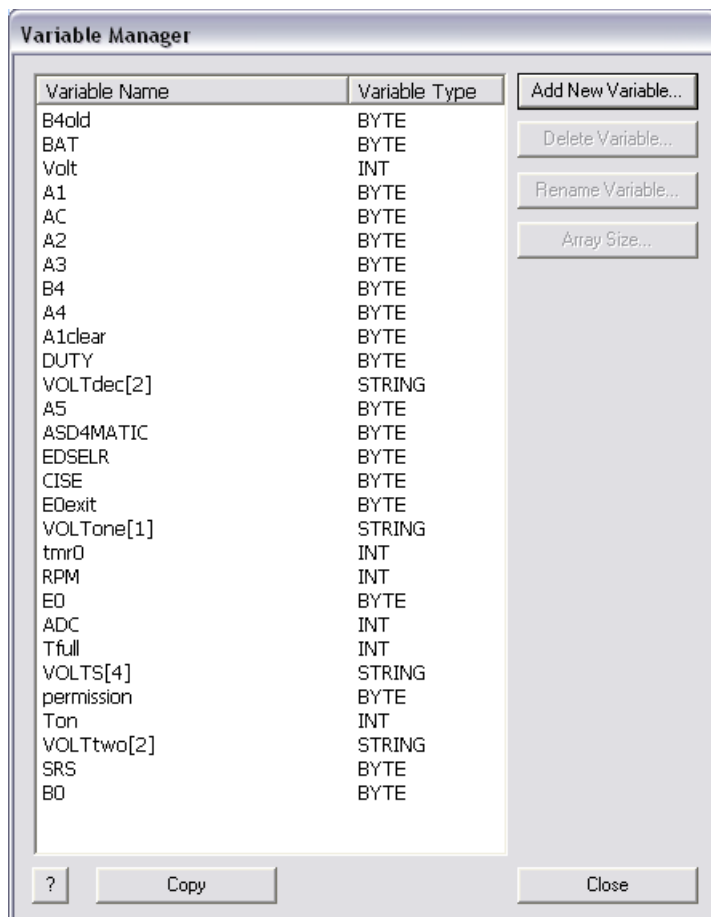
8.5 Ohjelma

Vikakoodinlukijan ohjelma tehtiin Flowcode V4:llä. Flowcodessa valitaan kontrolleriin kytketyt laitteet
ohjelman ylärivistä, tässä tapauksessa LCD, ADC ja painonapit. Nämä ilmestyvät paneeliin, joista nii-
tä voi käyttää simulaation aikana.



Kuva 28. Flowcode paneeli

Ohjelman sivupaneelista löytyy komponentit I/O:n määrittelyyn, keskeytyksiin, viiveisiin ja kytkettyihin laitteisiin. Työssä on käytetty kolmea eri muuttujatyyppiä, jotka ovat: byte (0-255), int (-32768-32767) ja string. String muuttujien koon voi valita tarpeen mukaan lisäämällä sen muuttujaan perään hakasuluilla.



Kuva 29. Kuvakaappaus muuttujista

Ohjelman ollessa valmis se käännetään HEX-tiedostoksi, joka ladataan kontrolleriin. Ohjelmointialustana käytössä oli alta 20€ maksanut, Ebayn kautta ostettu USB-pohjainen laite, jonka mukana tullut Micropro -ohjelma suorittaa ohjelman lataamiseen kontrolleriin.



Kuva 30. Ohjelmointilaite

9 KOTELOINTI

Koteloksi valikoitui Hammondin valmistama valualumiinikotelo 1590R1. Valualumiini on kestävä, helppo työstää ja on osoittanut laadukkuutensa aikaisemmissa projekteissa. Kotelon ulkomitat ovat 62,75mm x 192mm x 111,15mm (Hammond). Kotelon tärkein tehtävä on suojata laitteen herkkää elektroniikkaa mekaaniselta rasitukselta, jota se joutuu autokäytössä sietämään.



Kuva 31. Kotelo muoveissa

9.1 Kotelon työstö

Kotelo teipattiin maalarinteipillä merkintöjen tekemisen helpottamiseksi. Tällöin voidaan käyttää kuu-lakärkikynää permanenttitussin sijaan ja merkinnöistä saadaan tarkkoja. Mikäli mitoissa on virheitä, voidaan merkinnät poistaa helposti irroittamalla teippi ja teippaamalla uudestaan. Teippaus myös suojaa koteloä naarmuilta, jolloin pintakäsittely helpottuu. Näytön ja DB9-liittimen vaatimat isot rei-ät tein pistosahalla ja viimeistelin viilalla.



Kuva 32. Kotelo työn alla

Ensiksi hahmottelin koteloön näytön ja kytkimien paikat. Näytön vaatima aukko on 28mm x 83mm ja kytkimien reikien halkaisija 13 millimetriä. Koska käytössä oli hieman alkeelliset työkalut, valmistin ensin reiän näytölle ja vasta sen jälkeen näytön kiinnitysreikien paikat. Näin saadaan työstössä ta-

pahtuneet mittavirheet kompensoitua muuttamalla kiinnitysreikien sijaintia. Seuraavaksi tein virtajohdon vedonpoistimen ja DB9-liittimen paikat samalla tavoin.

Näyttö on korotettu kotelon kannesta neljällä M3-kierteisellä, 10 millimetriä korkealla jatkomutterilla. Näyttö on siis kotelon yläpinnan alapuolella, jotta laitteen osuminen esimerkiksi maahan ei särkisi näyttöä. Piirilevy on kiinnitetty niin ikään kahdella 30 millimetrin, M3-kierteisellä jatkomutterilla. Kuvassa 33 näkyy jatkomutterit ja painonappien kiinnitys.



Kuva 33. Kasausvaihe

En tullut huomioineeksi kotelossa olevien pohjakannen kiinnitysruuvien viemää tilaa sisämitoista, joten jouduin rälläköimään kaksi kierrekoteloita pois. Tästä ei kuitenkaan koitunut mitään suurempaa haittaa, sillä kansi kestää kiinni tukevasti neljälläkin ruuvilla kuuden sijaan. Johdotuksen näyttöön ja DB9-liittimeen olisi voinut kehittää paremmaksi.



Kuva 34. Laite kasattuna

9.2 Pintakäsittely

Teippauksen ansiosta kotelon pinta oli hyvässä kunnossa vielä työstämisen jälkeenkin. Näin ollen en nähnyt tarpeelliseksi maalata koteloita. Kotelon päälle tulostin tarrakalvon, jossa on kytkimien nimet, Mercedes-Benzin logo ja teksti scanner havainnollistamassa laitteen tarkoitusta. Kalvon asettelut tein Microsoft Officen Wordilla. Tekstit ovat tehty BankGothic-fontilla ja painonappien kirjaimet fonttikoolla 20 ja teksti scanner koolla 36. Koska Word ei ole tällaiseen käyttöön suunniteltu, jouduin tu-

lostamaan hahmotelman aina paperille ja vertaamaan koteloon. Kolmannella tulostuskerralla asettelu sattui kohdalleen.

E0 **A5** **A4**

A3 **A2** **A1**



SCANNER

Kuva 35. Kuvakaappaus kalvon asettelusta

Tarrakalvo on helppokäyttöinen: laitetaan kalvo tulostimeen ja valitaan tulostimen asetuksista tyypiksi kalvo. Tämän jälkeen tulostetaan kalvo, liotetaan vedessä kunnes kalvo irtoaa paperista ja asetellaan paikoilleen. Sitten kotelon annetaan kuivua ja lakataan. Lakkasin koko kotelon Mastonin spray -mattalakalla. Ainoa haastava osio on kalvon asettelu koteloon, sillä kalvo repeää herkästi ja kaikki lika reagoi lakan kanssa. Tässä epäonnistuin, sillä siirsin kalvoa useaan otteeseen ja maalasin lakan ennenkuin kaikki kosteus oli haihtunut. Tästä seurasi pieniä paakkuja kalvossa ja teksti meni osittain suttuiseksi.



Kuva 36. Laite toiminnassa

10 PÄÄTELMÄT

Opinnäytetyö oli kaikin puolin haastava projekti. Tietoa Mercedes-Benzin vikadiagnostiikan väylistä oli aluksi hankala löytää, sillä kaikki tuntui johtavan katsastuksessa suoritettavaan OBD-mittaukseen. Elektroniikan ja piirilevyn suunnittelun aloitin useasti puhtaalta pöydältä, kun järkevää ratkaisua ei syntynyt tai piirilevy meni liian monimutkaiseksi. Ohjelman tekoon ja muutoksiin meni projektissa kaikista eniten aikaa. Ohjelman lataaminen ICSP-väylän kautta ei myöskään toiminut, joten piirin jatkuva iroittelu kannastaan tuhosi muutaman kontrollerin.

Vikakoodinlukijan suunnittelu ja toteuttaminen sitoi elektroniikan ja sulautettujen järjestelmien kursien aiheita ja teorioita käytäntöön.

Insinööriyössä toteutettu vikakoodinlukija on täydellisesti toimiessaan erittäin hyödyllinen sähköisten järjestelmien vikojen etsintään ja bensiinin ruiskutuksella varustettujen moottorin arvojen seuraamiseen. Kehitän vikakoodinlukijaa ainakin harrastemielessä eteenpäin, kunhan olen saanut sen osa-alueet toimimaan nykyisessä muodossaan.

LÄHTEET JA TUOTETUT AINEISTO

Fault Code Manual for Mercedes-Benz. Baum Tools Unlimited Inc. 2001. [viitattu 26.5.2013]. Saatavissa: http://www.mbslk.com/files/artikel/r170_fehlercodes.pdf

HAIKO, Timo. 2013. Analoginen Elektroniikka. Helsinki: Sanoma Pro Oy.

Hammond 1590R1 – datasheet. [viitattu 24.5.2013]. Saatavissa: <http://www.hammondmfg.com/pdf/1590R1.pdf>

LYYRA, Pasi 2004. Käyttöohjeiden suunnittelu ja laatiminen piirilevyn valmistusprosessille. Mikkelin ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. [viitattu 24.5.2013].

MBCluster. [verkkójulkaisu]. Saatavissa: http://www.mbcluster.com/Media_Diagnostics/8-Pole.jpg

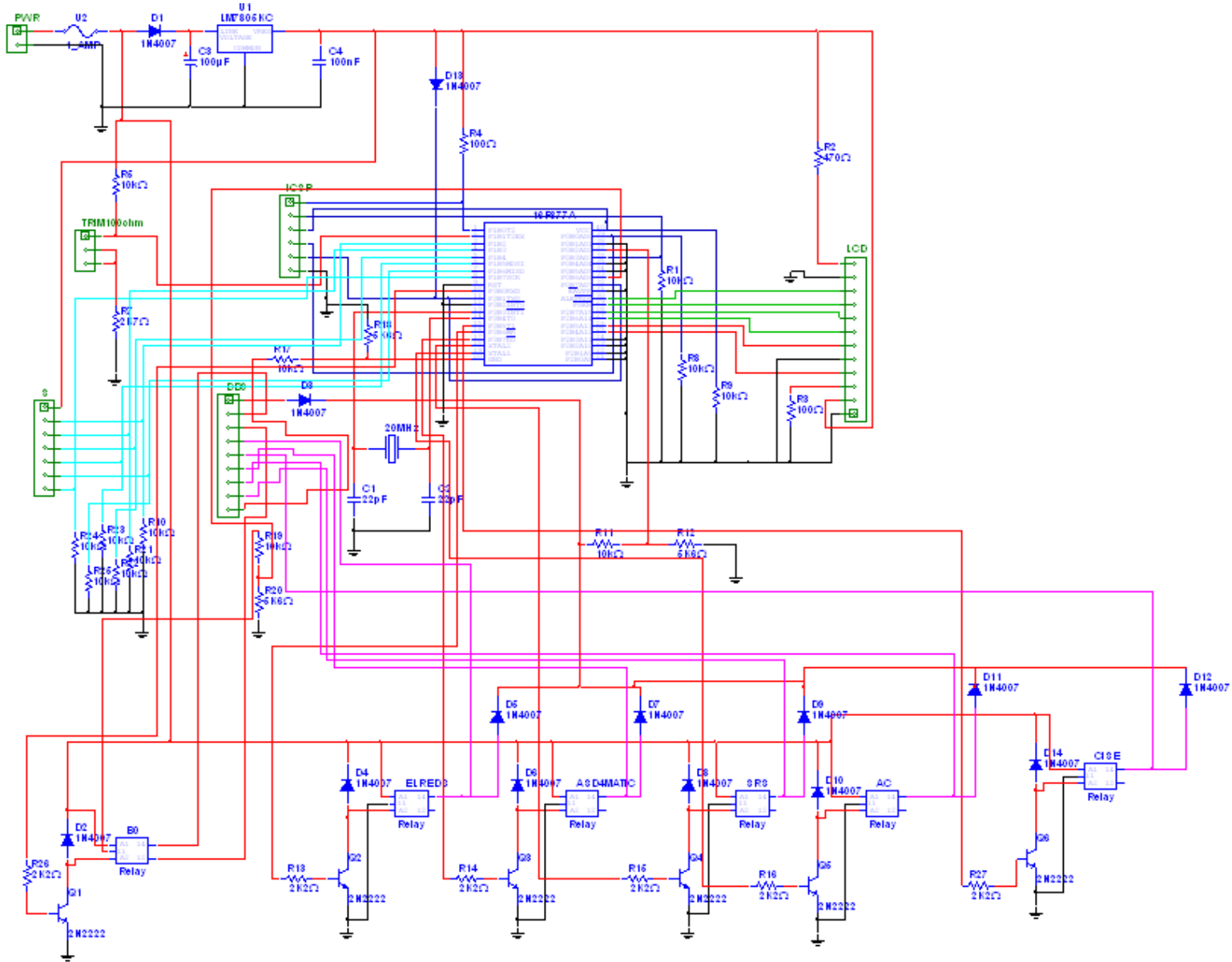
MicroChip PIC16F87XA-datasheet. [viitattu 29.5.2013]. Saatavissa: <http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>

STMicroelectronics 2N2222A –datasheet. [viitattu 28.5.2013]. <https://engineering.purdue.edu/ME588/SpecSheets/2n2222a.pdf>

WHITE, Charles 1998. Automotive Diagnostic Fault Codes Manual. Haynes.

LIITE 1

KYTKENTÄKAAVIO



LIITE 2

OSALUETTELO

Vastukset		Diodit		Transistorit	
R1	10k	D1	1N4007	Q1	2N2222
R2	470	D2	1N4007	Q2	2N2222
R3	100	D3	1N4007	Q3	2N2222
R4	100	D4	1N4007	Q4	2N2222
R5	10k	D5	1N4007	Q5	2N2222
R6	ei käytössä	D6	1N4007	Q6	2N2222
R7	2k7	D7	1N4007		
R8	10k	D8	1N4007		
R9	10k	D9	1N4007	Kondensaattorit	
R10	10k	D10	1N4007	C1	22pF kerko
R11	10k	D11	1N4007	C2	22pF kerko
R12	5k6	D12	1N4007	C3	100µF elko
R13	2k2	D13	1N4007	C4	100nF polko
R14	2k2	D14	1N4007		
R15	2k2				
R16	2k2				
R17	10k	Releet		Muut	
R18	5k6	RL1	SRD-12VDC-SI-C	U1	7805
R19	10k	RL2	SRD-12VDC-SI-C	Kide	20MHz
R20	5k6	RL3	SRD-12VDC-SI-C	MCU	16F877A HDD47780 4x20
R21	10k	RL4	SRD-12VDC-SI-C	LCD	LCD 6kpl lukkiutumaton
R22	10k	RL5	SRD-12VDC-SI-C	Painikkeet	2kpl
R23	10k	RL6	SRD-12VDC-SI-C	Haeunleuat	Vedonpoistin Hammond
R24	10k				1590R1
R25	10k			Kotelo	
R26	2k2				
R27	2k2			Liittimet	
TRIM	100 trimmeri			DB9 naaras	
				DB9 uros 2kpl	
				Hauenleuat 2kpl	
				4mm banaaniakki	
				5kpl	
				2,5mm monoplugi	
				4kpl	

PULSSIMUOTOISEN VÄYLÄN VIKAKOODIT (FAULT CODE MANUAL FOR MERCEDES-BENZ)

ASD/4matic

- 1 No fault
- 2 ASD computer broken / 4MATIC control module
- 3 Break light switch broken or wire interrupted
- 4 No revolution signal from front left wheel
- 5 No revolution signal from front right wheel
- 6 No revolution signal from rear axle
- 7 No revolution signal at all
- 8 Magnetic switch or break light switch broken / Over volts protection relay, front axle train valve (4matic)
- 9 Over volts protection relay, central differential lock valve (4matic)
- 10 Over volts protection relay, stop lamp switch, Rear axle differential lock valve (4matic)
- 11 Steering angle sensor signal (4matic)

ELR 201.126 1989

1. No fault
2. Speed sensor signal
3. Coolant temperature sensor
4. ELR control unit / Idle speed control ISC system

EDS

1. No fault
2. Fuel rack position sensor (L7)
3. Air flow sensor signal (B2/1)
4. Electronic diesel system (EDS) control unit (N39) / Atmospheric pressure sensor
5. Exhaust gas recirculation (EGR) valve vacuum transducer (Y31/1) / Fault in exhaust gas recirculation (EGR) control circuit
6. Electronic diesel system (EDS) control unit (N39), internal voltage supply
7. Starter ring gear speed sensor (L3)
8. Engine coolant temperature sensor (B11/4)
9. Intake air temperature sensor (B2/1a)
10. Voltage supply insufficient
11. Electronic idle speed control actuator (Y22) / Exhaust gas recirculation (EGR) valve vacuum transducer (Y31/1) / Boost pressure cut-out switchover valve
12. Not used
13. Electronic diesel system control unit (N39), faulty (internal fault memory)
14. Electronic diesel system pressure sensor (B5/1), defective

15. Boost pressure control/pressure control flap vacuum transducer (Y31/5) / defect in boost pressure control circuit / Intake manifold air pressure control valve vacuum transducer (Y31/2), wastage vacuum transducer (Y31/3), or malfunction intake manifold air pressure circuit

SRS 107 126 201 140

- 1 No fault found
- 2 SRS Control unit
- 3 SRS - Driver air bag
- 4 Front passenger Airbag
- 5 Driver seat belt buckle
- 6 Front passenger seat belt buckle
- 7 Airbag resistor, Front passenger
- 8 Circuit 15R, Voltage supply
- 9 Warning lamp faulty
- 10 Control unit was activated

SRS 124 129

- 1 No fault found
- 2 SRS Control unit self test failure
- 3 Driver Airbag squib
- 4 Front passenger Airbag squib
- 5 Airbag/ETR, Driver seat belt buckle switch
- 6 Front passenger seat belt buckle switch (ETR)
- 7 Front passenger Airbag resistor
- 8 Voltage supply interrupted
- 9 SRS Warning Lamp (with flashing SRS warning lamp Impulse counter scan tool button held too little time to read out the DTC memory or too long to erase DTC codes. Reread codes.)
- 10 SRS Control unit activated.

A/C

- 1 No fault found
- 2 Fuel pump relay (circuit 87) not functioning
- 3 TN/TD signal (RPM) interrupted
- 4 Output for oxygen sensor heater control defective
- 5 Output for air injection pump control defective
- 6 Output for kickdown switch control defective
- 7 Not used
- 8 Engine coolant temperature sensor signal out of range
- 9 Circuit 50 failure
- 10 Output failure of the start valve
- 11 A/C compressor engagement signal missing (87Z)
- 12 Output for A/C compressor control defective
- 13 Excessive A/C compressor clutch slippage
- 14 Vehicle speed signal illogical
- 15 Short circuit detected in fuel pump circuit

